CIRCUIT MODELLER FOR CP/M

By Harcourt Systems.

DISTRIBUTED BY

SEASIM ENGINEERING SOFTWARE
THE PADDOCKS, FRITH LANE,
MILL HILL, LONDON NW7 1PS.

Tel: 01-346-9271 Telex: 28915

CIRCUIT MODELLER

For the first time - a CAD package for CP/M™* microcomputers for electrical and electronic circuit design.

An essential tool for electrical/electronics designers, Circuit Modeller solves linear electrical/electronic circuits for DC, LF, HF, VHF and UHF and plots graphs of response.

The basis of use of the package is to solve a circuit on paper before committing to hardware building and testing. Since "cut-and-try" methods form a part of almost all designs when a computer is not used, Circuit Modeller almost automatically cuts costs by speeding the design engineer's job. Also designs can be more accurate, because component imperfections (linear ones that is) can all be included - there are fewer approximations.

Circuit Modeller in use

Gain/loss, phase angle, and complex Zin and Zout can all be listed in a table on the screen or on the printer for chosen frequencies. Here is a typical printout of gain and phase for a circuit where the user types in each frequency required:

```
 FREQUENCY (HZ)     GAIN                    PHASE


  100            -47.3545  dB          174.764  DEGREES
  1000           -7.99469  dB          122.044  DEGREES
  2000           -1.15894  dB          68.5823  DEGREES
  3000           -.220992  dB          43.7305  DEGREES
  5000           -.015382  dB          25.1153  DEGREES
  10000           2.02969E-03   dB     12.2426  DEGREES
```

And here is one of input and output impedances:

```
FREQUENCY (HZ)   ZIN - Real          Imag.        ZOUT - Real                Imag.
 2000                 48908.9 , j -18750.2              .0074876 , j  .0298584
 3000                 55834.7 , j -7456.62              .0138239 , j  .035209
 4000                 59938.8 , j -3361.63              .0160581 , j  .0406319
 5000                 62383.4 , j -1637.25              .0170155 , j  .0469062
 10000                66427.8 , j -15.1006              .018212 , j  .0832771
```

Another facility of Circuit Modeller is to find bandwidth. BW finds the frequency of maximum gain, (the centre frequency), searches for the upper and lower frequency limits at which the voltage gain/loss is 3 dB down, and then prints the three frequencies, and the bandwidth.

SEARCH takes a given gain/loss from the user and finds the frequency at which this gain/loss occurs. This is useful for finding the cutoff frequency of a circuit, or the frequency at which the gain is 0dB so that the phase margin can be found.

Circuit Elements

Circuit elements which can be included in models are as follows:

- R   Resistances, can be specified in ohms or Kohms.
- C   Capacitances, can be specified as µF, nF or pF.
- L   Inductances, can be specified as µH, mH or H.
- MU   A pair of mutual inductances with L1, L2 and M.
- VS   A voltage controlled voltage source.
- CS   A voltage controlled current source used for hybrid-pi.
- OA   An operational amplifier.

OA is interesting because it models the operational amplifier to include the effects of the first pole. Also it models the input and output resistances and the open loop voltage gain. Six different operational amplifiers are included in the library, and the user can specify alternatives.

The capacity of Circuit Modeller is:

- 101 resistors
- 101 capacitors
- 101 inductors
- 31 mutual inductors
- 31 voltage sources
- 31 current sources
- 31 operational amplifiers

However the memory limits the number of nodes to 32, so it is unlikely that all of the above would be used at once.

General

Circuit models can be saved on the disk and loaded again later.

Processing speed is enhanced in the all-compiled package by optimally coded inner loops using macro-assembler. This cuts the solution time for an example eight transistor wideband amplifier to 6 seconds, (8 hybrid-pi models in cascade) instead of the six minutes for an ordinary BASIC program.

An automatic plotting function (PLOT) is available as an optional extra. This uses a normal printer capable of being switched to 132 columns to plot the dB gain/loss and the phase shift on the same graph against log frequency, with automatic axis scaling. The user chooses any frequency range of interest from 0.1 Hz to hundreds of GHz. Several examples of PLOT output are shown on the next two pages.

System Requirements

A Z80 microcomputer supporting CP/M™* is required with at least 48,500 bytes of RAM memory free for Circuit Modeller. Allowing for your CP/M™* this requires at least 60 KBytes of RAM total. An exception to this rule is the RML 380Z which requires 56KBytes RAM. If the PRINT and PLOT features are required then a printer capable of being switched to (or normally possessing) 132 columns is needed. Most 15 inch dot-matrix printers and most daisy-wheel printers are suitable. Also the Epson MX80 printer can be used.

Circuit Modeller will run with only one disk drive if required, provided that at least 90 KBytes can be accommodated on the drive.

Support Documentation

Two manuals are supplied with Circuit Modeller. The Operating manual contains instructions for operation of the program, and the theory manual gives details of how models can be made and elementary examples of the use of Circuit Modeller.

Machines for which CIRCUIT MODELLER is available.

Machines currently supported are Z80 based with the following diskette drive formats:

Any 8 inch IBM 3740 single sided single density diskette drive Z80 CP/M™* machine.

(The above 8 inch format covers about 30 different types of computer)

Superbrain 35 track 5.25 inch SS/DD
Cromemco 5.25 & 8 inch (CP/M)
Tuscan 5.25 SS/DD, 8 inch SS/SD
RML 380Z, MDS (CP/M, Varitext and COS4.0 required)
Sharp MZ80B

If your diskette format is not listed, then you may obtain an 8 inch IBM diskette from us, and have your dealer make you the right diskette for your machine. Please note that the ICL and Rair are unsuitable, (as is any non-Z80 machine) and the Osborne 1 and Xerox have too small a disk size for Circuit Modeller. The new Osborne (Double Density) has only 52 columns showing on the screen at once, and Circuit Modeller uses 80 columns. Otherwise it is suitable if you can obtain disk conversion from your dealer.

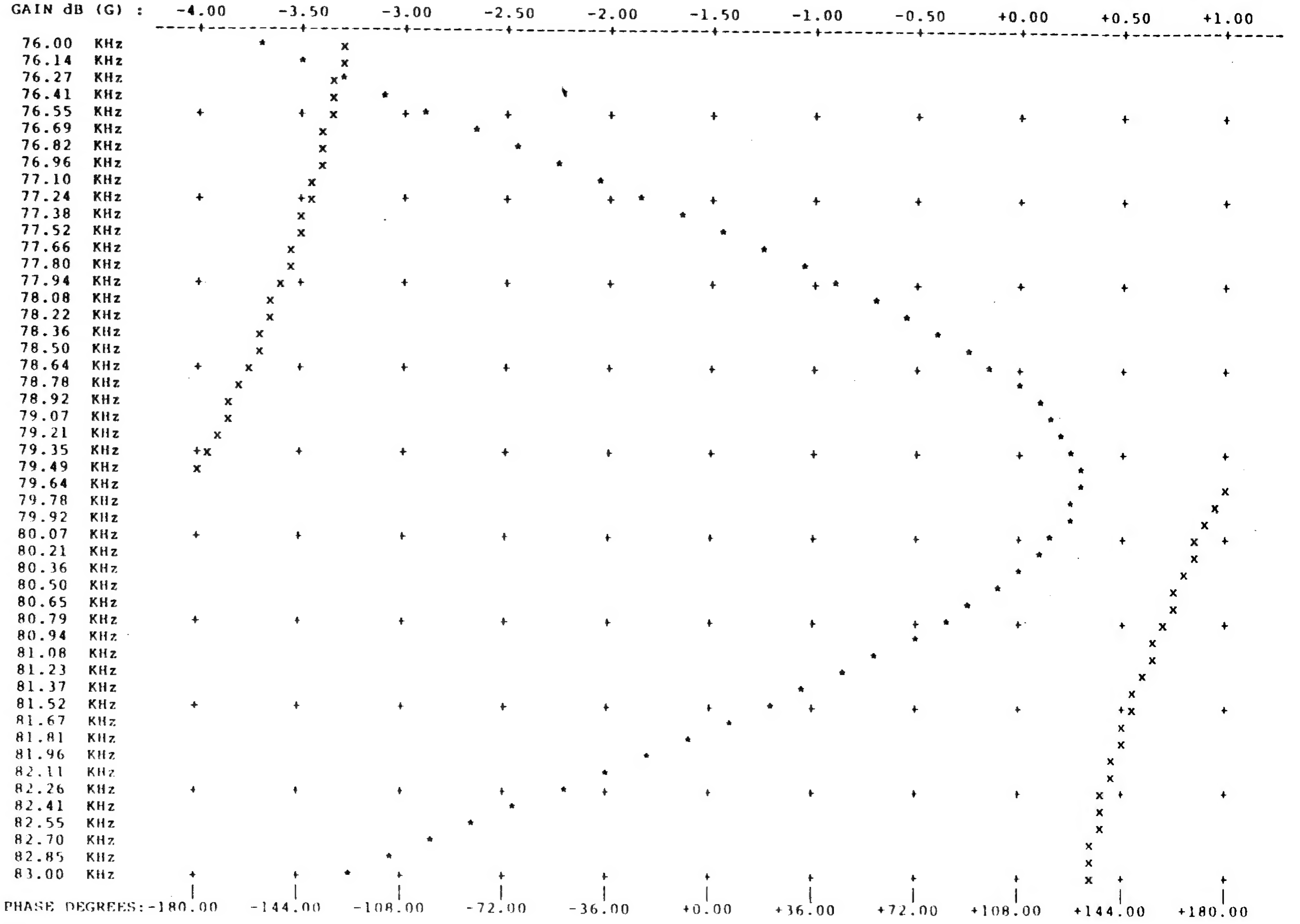
Availability.

Circuit Modeller is available from the address below at £125.00 inc. and the manuals are available separately at £10.00 inc.
The optional PLOT feature is an extra £35.00. Users may upgrade to include PLOT at a later date.
VAT is chargeable at 15%.

*CP/M is a trademark of Digital Research Inc.

Seasim Controls Ltd., The Paddocks, Frith Lane, Mill Hill
LONDON NW7 1PS Tel: 01-346-9271 Telex: 28915

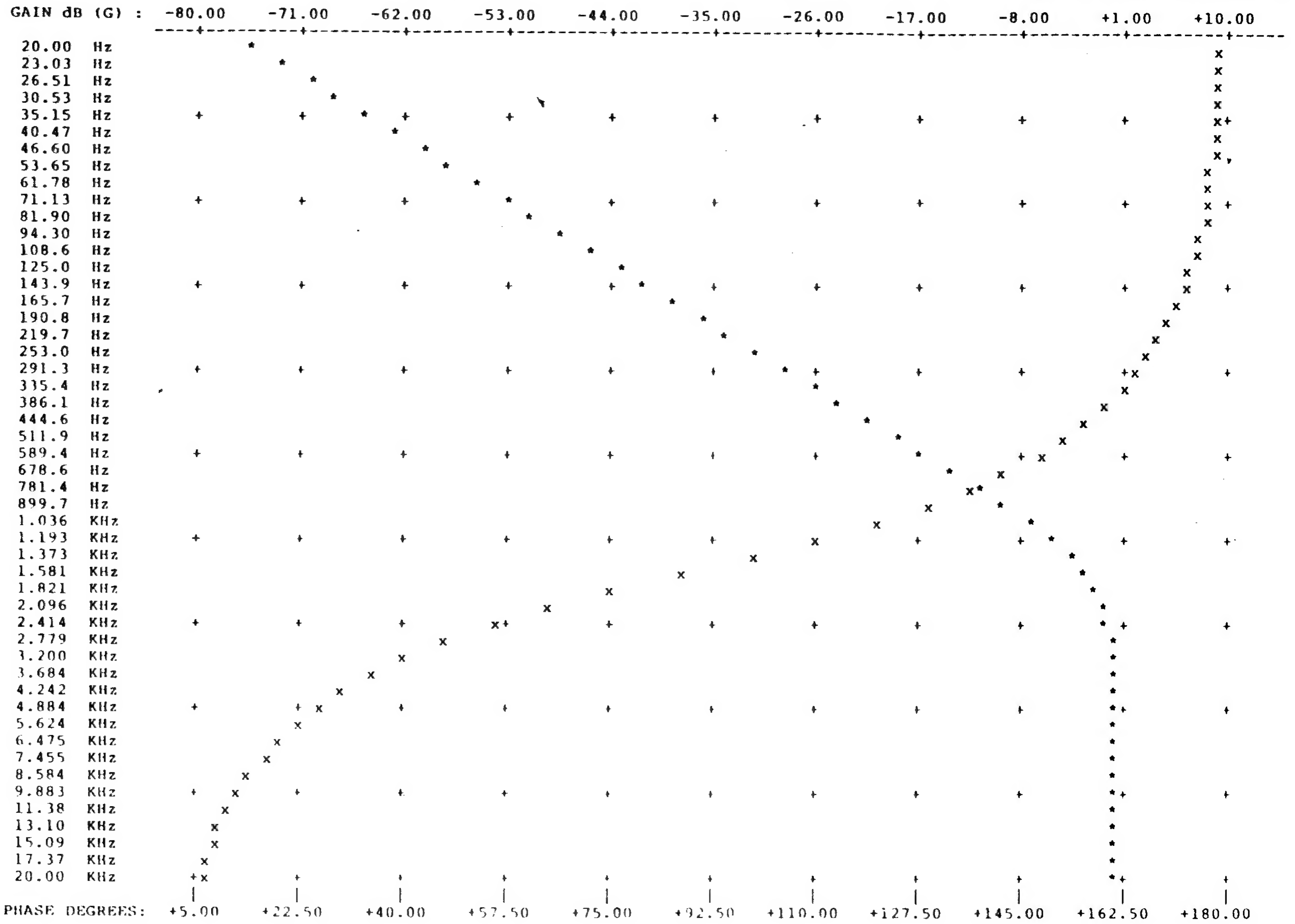
```
This copy of Circuit Modeller is licenced to HARCOURT SYSTEMS
FREQUENCY RESPONSE CURVE  OF FETOSC -- Demonstration of an oscillator design.
                                                                    KEY: x PHASE * GAIN # BOTH
```



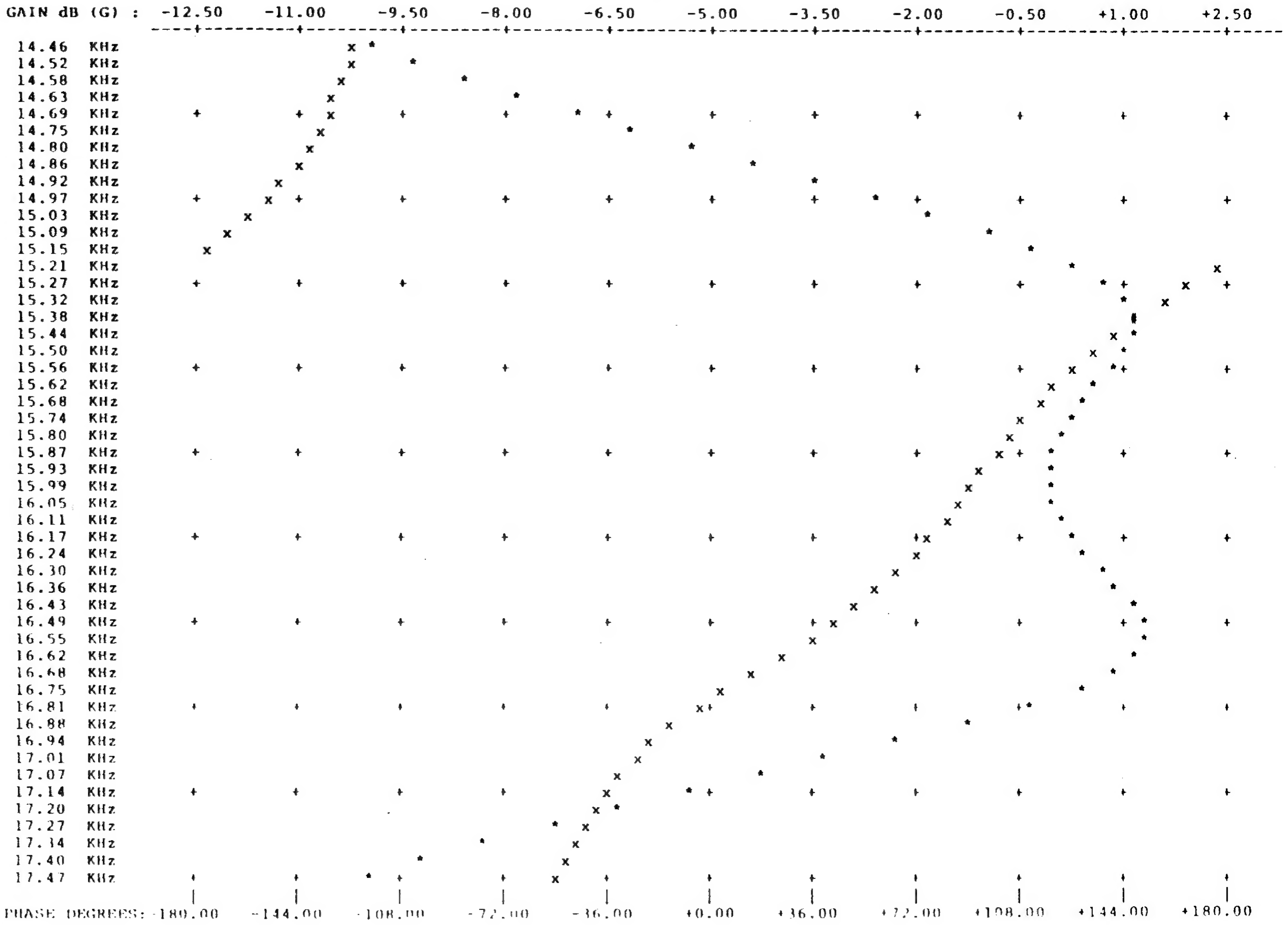
SEASIM ENGINEERING SOFTWARE

This copy of Circuit Modeller is licenced to HARCOURT SYSTEMS
FREQUENCY RESPONSE CURVE OF TEST -- Demonstration of an active filter design.

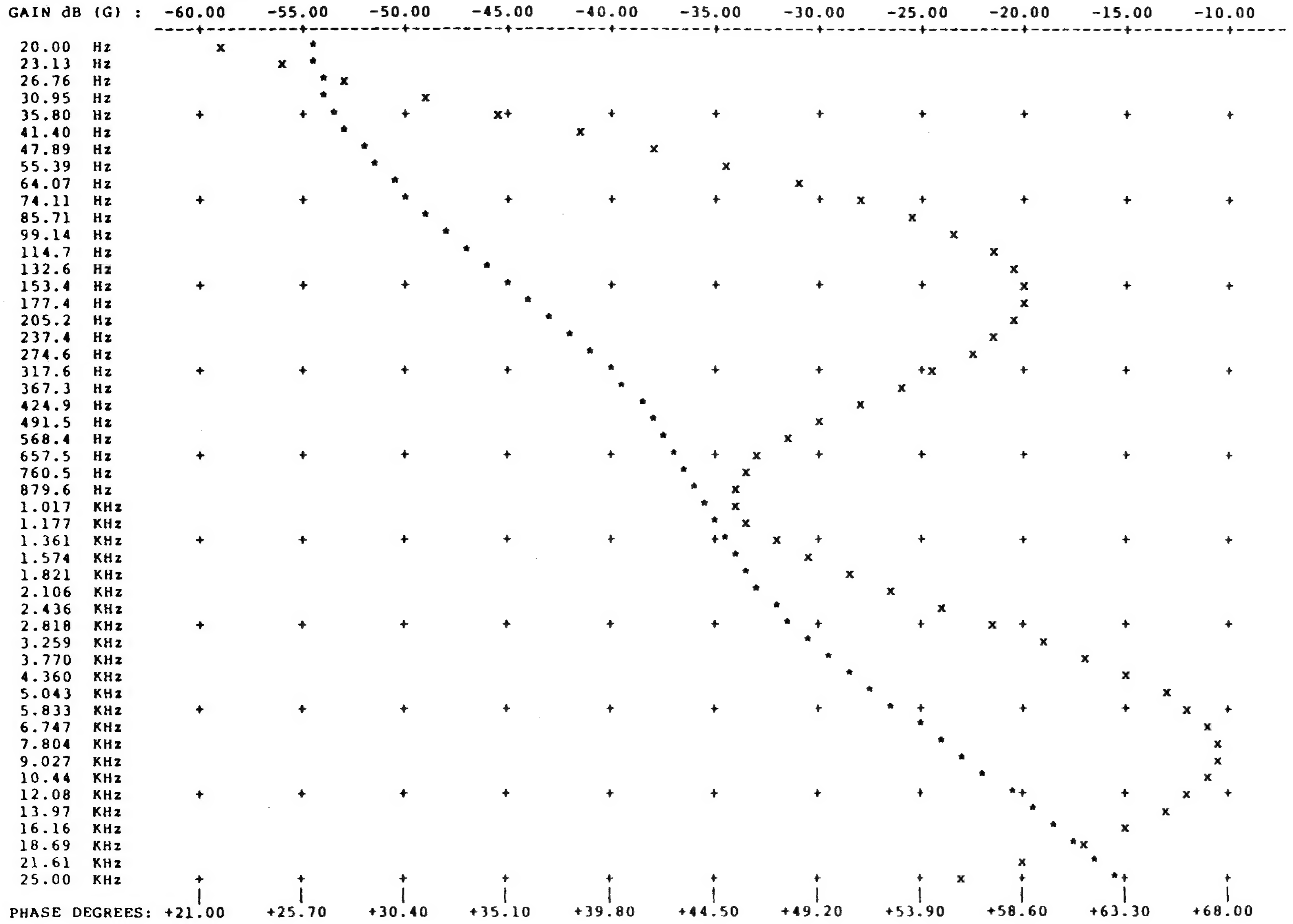
KEY: x PHASE * GAIN # BOTH



This copy of Circuit Modeller is licenced to HARCOURT SYSTEMS
FREQUENCY RESPONSE CURVE OF COUPLED -- Demonstration of an IF Coupling Transformer design. KEY: x PHASE * GAIN # BOTH



```
This copy of Circuit Modeller is licenced to HARCOURT SYSTEMS
FREQUENCY RESPONSE CURVE  OF RIAA -- Demonstration of an (RIAA) filter design.          KEY: x PHASE * GAIN # BOTH
```



SEASIM ENGINEERING SOFTWARE

CIRCUIT MODELLER FOR CP/M

Order Form

COMPUTER DETAILS

MAKE:
APPROX.DATE SUPPLIED:
NAME & TEL.NO. OF SUPPLIER:

NOTE: Only Z80 based Computers with a minimum of 60K bytes RAM are suitable. Please see information sheet.

If in doubt, please contact us or your supplier.

DISK DETAILS

If integral with Computer, tick box □
and ignore further Disk questions.
If separate, please complete:

MAKE:
APPROX.DATE SUPPLIED:
NAME & TEL.NO.OF SUPPLIER:

APPROX.CAPACITY PER DISK SIDE: □
MINI FLOPPIES ($5^1/_4$"): □
FLOPPIES (8"): □
DOUBLE SIDED: □
DOUBLE DENSITY: □

CIRCUIT MODELLER OPTIONS	QUANTITY	PLOT (requires 132 column printer)	QUANTITY	TOTAL
SINGLE PRECISION @ £143.74 inc. VAT		£40.25 inc. VAT		
DOUBLE PRECISION @ £143.74 inc. VAT		£40.25 inc. VAT		
BOTH VERSIONS @ £216.20 inc. VAT		£60.95 inc. VAT		
MANUAL ONLY (supplied with Disks) £11.50 inc.VAT		Postage & Packing £1.00 Air/Surface Mail at Cost.		

CREDIT CARD NO:

ACCESS:

BARCLAYCARD:

Please make all cheques and credit cards payable to: SEASIM ENGINEERING SOFTWARE.

How did you hear about CIRCUIT MODELLER? ______________________

What other sort of Engineering software would you like us to make available? ______________________

Name: Mr/Ms ______________________

Address to which goods are to be sent ______________________
____
____

Please allow 21 days delivery.

DISTRIBUTED BY: SEASIM ENGINEERING SOFTWARE,
THE PADDOCKS, FRITH LANE,
MILL HILL, LONDON, NW7 1PS.
Telephone: (01) 346-9271
Telex: 28915

VAT Reg.No. 230 5788 64

Seawave Simulation®

Seasim Controls Limited

The Paddocks, Frith Lane
Mill Hill, London N.W.7 1PS

Telephone: 01-346 9271
Telex: 28915
Cables: Seawave London

Date as postmark

Dear Enquirer,

Thank you for your interest in Circuit Modeller. We have pleasure in enclosing a temporary brochure explaining some of its features.

We are sure that you will recognise the importance of this new engineering program for CP/M* Computers. We believe it will not only save you valuable time but also enable you to explore new designs that would otherwise have been impracticable.

You will notice that your enquiry is being dealt with by SEASIM ENGINEERING SOFTWARE, which under an agreement with Harcourt Systems is now distributing Circuit Modeller. Seasim itself is involved in high technology engineering, including the design of electronic equipment and it is because Seasim have recognised the importance of Circuit Modeller that we are now making this available to you.

Yours sincerely,

M. Wakefield.

Secretary

* CP/M is a trade mark of Digital Research.

Registration Form

Tatum Labs
P.O. Box 722
Hawleyville, CT 06440
Phone 203-426-2184

Name:

Company:

Address:

City, State, Zip:

Purchased from:

Program:

Serial Number: Date of purchase:

Computer:

Number of disk drives: Ram size:

Operating system:

How did you find us?

What other programs would you like us to make?

Comments:

========================Fold, stamp and mail to:==========================

Place
Stamp
Here

Tatum Labs
P.O. Box 722
Hawleyville, CT 06440

Electronic Circuit Analysis -- addendum 01/14/83

Two utilities were added since the manual was printed:
RET.COM and PLOT.BAS

RET.COM

This will return to ECA after leaving by the "Q" command. All data remain intact. Do not use after running any other program. The intention is twofold. It allows the use of the CP/M built-in commands (era, dir, ren, save, type, user). Second, it provides a means of recovery from Bdos errors, which cause a warm boot.

It will have unpredictable results if the transient program area has been disturbed, so do not run any programs. It may or may not work after a cold start, because some cold start loaders change the TPA.

We are constantly improving the program. At any time, if we haven't raised the price, you may send your disk back with return postage for a free upgrade to the latest version.

PLOT.BAS

This is a plotting utility for plotting results from an AC analysis. It requires Microsoft Basic to run. (It will run on either version 4 or 5.) It will produce a graphic plot of amplitude or phase against frequency. It will print on any ordinary printer. It is self prompting and will ask for all the information it needs. It requires a file generated by the "SD" command.

The text inside parenthesis after each request is the default. Normally, the defaults are what was specified on the last plot. This enables you to look at several possible arrangements on the screen before printing it on paper.

It does not have auto scaling. The questions "Upper limit" and "Lower limit" refer to the limits of the plot. They are not necessarily the limits of the data. Data outside the limits will be plotted at the limits.

The question "Divisions" asks how many divisions you want the scale divided into. You should pick a number that results in a convenient division size.

The program will automatically label the divisions, where it can. It will label the end points and some or all of the in-between points, depending on what fits. It may choose different in-between points for printer and screen if they are not the same size.

This is a primitive graphics package. No attempt was made to support any high resolution graphics, due to the wide variation in available equipment.

There are 3 points where most users will want to customize it.

Display (CRT) line length is variable "CU" defined in line 30.

Printer line length is variable "PU" defined in line 40

A string "PI$", defined in line 50, will be sent to the printer at the start of each plot to initialize the printer. As supplied, no string is sent. The string in line 50 sets the Epson MX-80 printer to.condensed characters for a 132 character line. As supplied, there is an apostrophe at the begining of this line to eliminate its effect. If you have an Epson printer, simply remove this. If you have any other printer, this is where to define the string to set it to the desired character size.

ECA.BAS

ECA.BAS is a circuit analysis program written in Basic. It requires Microsoft Basic (either version 4 or 5) to run. Its features are a subset of ECA.COM, which was not written in Basic.

Most of the commands are the same. The following commands are not implemented:

C, D, E, F, G, I, K, O, T, W

The only options implemented are D and L.

The printer is turned on by the command "P" instead of "OL". The output nodes are set by the command "N". "N" will display the present output nodes, and ask for a node number. Hit enter to keep the old one, otherwise specify a number. The "H" command displays a menu.

On sign on, it will ask for the maximum number of nodes and the maximum number of branches. You may specify any number equal to or larger than the actual number used. Execution will be fastest if the smallest number that allows the desired circuit size is specified. Do not be alarmed at the long initialization time.

It will also ask single or double precision. If you select double precision, the prompt becomes "->>" instead of "-->". Double precision is much slower than single precision, but may be used to check the accuracy of the single precision results of the main program. The Basic version will run approximately 10 times slower than the machine language version, even though it is highly optimized. Double precision is still slower by a factor of 3 to 4.

Electronic
Circuit
Analysis

Tatum Labs
P.O. Box 722
Hawleyville, CT 06440
203-426-2184

Table of Contents

NOTICE

Electronic Circuit Analysis is warranted for one year to be free from coding defects. Should a problem arise, we will correct it at no charge during the warranty period. This service shall only be provided to purchasers who register within 30 days of purchase.

This warranty shall not extend in any way beyond correction of the program itself. Neither Tatum Labs nor any authorized dealer shall have any other liability or responsibility to any person or entity with respect to any loss or damage caused or alleged to be caused by this computer program, including but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such computer programs.

Good programming practices dictate that frequent backup copies be made to protect active files. Also, valued data should not be used under an unknown system until it has been thoroughly tested.

Introduction

Electronic Circuit Analysis is a program for microcomputers to analyze circuits. It is intended to provide an alternative to breadboarding and extensive measurements. The circuit can be "tweeked" on the computer without actually building it. A very high degree of accuracy is available. When a circuit is optimized on the computer, you are not compensating for component tolerances.

The program has many advanced features including a full worst case analysis, for finding variations in performance with component tolerance, and a dynamic modification facility, which allows component values to be varied by the computer, and a table of results to be printer automatically. By using control files, a series of operations can be done without operator attention.

This program can analyze circuits of up to 64 nodes and 127 branches. (48k or more free memory) Execution time will vary from less than one second per frequency to about one minute per frequency depending on the circuit size and complexity. It is rare that it will take more than 10 seconds.

Larger circuits, if they can be broken up into stages that fit this requirement can be analyzed using the "I" command. This command uses the results of one analysis as the input for the next. Using this method, circuits of infinite size can be analyzed.

This program will also perform a worst case analysis, for finding the production extremes of performance. Worst case information is generated for phase as well as amplitude. This is useful for finding shifts of filter center frequencies and similar information. The worst case flags are stored for later use, so a sweep can be done. It will also allow dynamic modification of the circuit, and automatic repeat analysis.

It contains a chaining facility so long multiple runs can be done without operator intervention. It contains full editing and error trapping routines, so the user will not have to re-enter the whole circuit because of any errors.

All circuits are built from passive components (resistors, capacitors, inductors) and active components (voltage and current sources, controlled and fixed). Models of other devices such as transistors, op-amps, and vaccuum tubes can be built using the components available in this program.

Some possible uses are analyzing filters, audio amplifiers, equalizers, antenna couplers, and many others. The computer can, in a matter of seconds, analyze circuits that were impractical to do by hand, to a much greater degree of accuracy.

Starting

To run this program, type and enter the dos command:

```
      ECA
```

Electronic Circuit Analysis will then load and execute. It will display the heading, copyright message, the highest allowed branch number, and the highest allowed node number and the prompt.

The prompt (-->) shows that the program is in the command mode. You should enter a command. Normally, the first command will be "B" to build a circuit. First time users should turn to the examples section for further assistance.

It is good practice to make a backup of the master disk before using the program, and then run only the backup.

This is a non-system disk. Single drive users must first transfer the files to a system disk.

Notation

Throughout this manual, the following notation conventions are used:

Upper case letters must be typed exactly as printed.
Lower case letters represent prototype values for which the user should substitute his own values.
Braces { } indicate optional parameters.

Building a circuit (B command)

The "B" command is used to build a circuit. The command takes the form:

```
      B{D}{b1{,b2}}
```

Parameters (all optional)

```
     D    =   use disk file
     b1   =   start branch    (default is 1)
     b2   =   end branch      (default is 127)
```

In normal use, you will just enter "B" to enter the circuit fom the keyboard, or "BD" to read the disk.

This command writes over the old values. Any old data not written over is retained. If it is necessary to start over, an "R" command my be executed to clear the list before building the circuit.

To exit this mode, push return with a blank line. Control-c or control-z will also work.

Simple components

Components are entered in the build mode in this form:
{bn,}t,n1,n2,v,{t1}

bn	=	branch number (optional)
t	=	component type
n1	=	first node number
n2	=	second node number
v	=	value
t1	=	tolerance (optional)

If a branch number is specified, the data is entered in to that branch, and the branch counter is changed to that number. If it is left out, the number displayed is used. This overrides any numbers given in the command.

The component code must one of the following letters:

C	capacitor (farads)
D	delete branch
G	conductance (mhos)
I	current source (see notes)
L	inductance (henrys)
R	resistance (ohms)
V	voltage source (see notes)

Any other character represents a blank branch.

The two nodes the component connects between are integers between 0 and 64. Node 0 is used as a reference for all calculations and is assumed to have a voltage of zero. (This is the ground, earth or common node)

The component value is a floating point number. Values must be entered in the units specified in the above list.

The tolerance in percent is entered next. If no tolerance is specified, it is assumed to be 0. It is only used in worst case analysis. The tolerance will be truncated to one decimal place (.1 %) and must not exceed 25 percent.

Typical entries:

R,3,4,10000 represents a resistor between nodes 3 and 4 with a value of 10 kilohms and no tolerance.

C,9,0,1E-6,5 represents a capacitor between node 9 and ground with a value of 1 microfarad and 5 % tolerance

5,G,23,45,.01 represents a conductance between nodes 23 and 45 with a conductance of .01 mhos, replacing branch 5. The next branch will be number 6.

8,D . . . . . means to delete branch 8, leaving it blank

D . . . . . . means to delete the current branch, or leave a blank branch

Sources

Sources ("I" and "V" codes) are entered as two branches.

```
     t,n1,n2,v1,{t1}
     R,n3,n4,v2,{t2}

     t   =  V = voltage source, I = current source
     n1  =  positive input node
     n2  =  negative input node
     n3  =  positive output node
     n4  =  negative output node
     v1  =  source value
     v2  =  resistance
     t1  =  source value tolerance
     t2  =  resistance tolerance
```

For a fixed source, make both input nodes (n1 and n2) zero. Otherwise the source is "controlled".

Source value units:

```
     fixed voltage   :  volts
     fixed current   :  amperes
     controlled voltage : voltage gain   ( v out / v in )
     controlled current : transconductance ( i out / v in )
```

The resistance specified as v2 is in series with voltage sources, or in parallel with current sources. It must be finite and not zero. Ideal sources are not allowed.

When numbering nodes, the node between the voltage source and its series resistor is not counted. All sources are converted to current sources internally.

If the second (R) branch is not present, an "Illegal math function" error will occur when the circuit is analyzed.

Typical entries:

```
V,0,0,1          represents a fixed voltage source of 1 volt
R,1,0,100        with a series resistance of 100 ohms.
                 It connects between node 1 and ground.
                 Node 1 is "in phase" (0 degrees).

V,1,2,10000      represents a voltage controlled voltage
R,3,4,50         source (amplifier) with a voltage gain of
                 10,000 and an output resistance of 50 ohms.
                 The input is diferentially connected between
                 nodes 1 and 2.  The output is connected
                 between nodes 3 and 4.
                 Node 3 is in phase with node 1.

V,2,0,1          represents a unity gain inverter with
R,0,7,5000,5     input at node 2 and output at node 7,
                 and a series resistance of 5000 ohms,
                 and a resistance tolerance of 5 %.
                 There is no tolerance on the gain

I,7,54,.05,25    represents a voltage controlled current
R,9,3,220E3,1    source with a transconductance of .05 mhos
                 (amps out per volts in)  and a shunt
                 resistance of 220 kilohms.
                 The input is between nodes 7 and 54.
                 The output is between nodes 3 and 9.
                 Node 9 is in phase with node 7.
                 The transconductance has a tolerance of 25%
                 and the resistance has a tolerance of 1%.
```

Listing the circuit (L command)

The "L" command lists the circuit in memory.

```
The "L" command lists the circuit in memory.
     L{opt}{b1{,b2}}
Parameters (optional)
     b1   =  start branch   (default is 1)
     b2   =  end branch     (default is 127)
Options:
     D    =  disk
     L    =  line printer
     X    =  inhibit route to disk
     S    =  inhibit line printer
```

The list is displayed on the terminal regardless of any options. Parameters may be combined for multiple destinations. For example, "LLD" sends the list to the disk, printer and screen simultaneously.

If the printer was enabled previously by the "OL" command, printing may be defeated by the "S" option. If spooling is active, LX will defeat the feed to the spool file.

The tolerance is not listed if it is zero.

Starting and ending branches may be specified as above.

Control-s will cause a pause in the listing. Any key will resume the list. Control-c will stop the list and return to command mode.

Saving the circuit (LD command)

The entire component list may be saved on a disk by entering "LD" (list to disk) at the command level. The file is in an ASCII format compatible with most word processors, so the list may be used as part of a report. All branches will reload in the same place regardless of numbers specified in the BD command. A partial list may be saved by entering "LD b1,b2". All branches from b1 to b2 will be saved.

The program will ask for the file name, then a heading. The heading is simply a line of text that will be saved at the beginning of the file.

The list will be simultaneously displayed on the screen in the same format as is being written on the disk or tape. This process may be stopped at any time by control-c. If this is done, the disk file may not be readable since the disk directory is not updated. This is recommended only if you realize you don't want to use the file.

If the file name specified already exists, the old file is deleted and replaced by a new file of the same name.

Modifying the circuit (M command)

The "M" command is used for a simple value or tolerance modification.

```
     M{T}b1{,b2}
Parameters:
     T    =  modify tolerance  (otherwise value)
     b1   =  start branch  (or only branch)
     b2   =  end branch  (optional)
```

The value or tolerance (only) may be examined and modified by the "M" command. A group of branches may be modified at the same time by specfying two numbers in the command. The command "Mb1" will cause the value (resistance, etc.) in branch b1 to be modified. The command "Mb1,b2" will cause all branches from b1 to b2 to be modified. "MT" will modify the tolerance instead of the value.

The program will display the current values, including type and nodes, and wait for an entry. If a number is entered, it will become the new value. If no number or a zero is entered, the old value will be retained. The new data is then displayed and the program steps to the next branch, or returns.

"MT" will modify the tolerance. The same conditions apply except that zero is a legitimate entry. Only a null will cause the old value to be saved.

You may break out of this mode at any time by entering control-c. All changes have already been entered.

Only the value or tolerance may be changed by this command. (One entry) The type and node numbers may not be changed this way. If it is necessary to change the node numbers or component type, use the "B" command.

Analyzing the circuit (A, S commands)

A simple analysis may be done by the "S" (sweep) command.

```
S{opt}f1{,f2{,st}}{:N=n1{,n2}{,...,nn}
```

A detailed analysis (all nodes) is done by the "A" command.

```
A{opt}f1{,f2{,st}}
```

Parameters:

f1	=	start frequency
f2	=	end frequency
st	=	step size if positive step multiplier if negative
n	=	node numbers (default is highest node)

Routing options:

L	=	data to line printer
D	=	data to disk file
S	=	inhibit line printer
X	=	inhibit disk

Analysis options:

A	=	use previous ":N" optional nodes
V	=	use previous worst case voltage flags
P	=	use previous worst case phase flags
+	=	use previous worst case increased flags
-	=	use previous worst case decreased flags

The command "Sf1,f2,st" performs a sweep from f1 Hz to f2 Hz, with a step size or multiplier of st, for the highest numbered node. A detailed analysis (all nodes) is done by the "A" command.

If st is positive, a linear sweep is done; and st represents the number added to each frequency to get the next. If st is negative, a logarithmic sweep is done; and st represents the multiplier. "S100,300,10" indicates a linear sweep from 100 Hz to 300 Hz in 10 Hz steps. "S20,20000,-2" indicates a logarithmic sweep from 20 Hz to 20000 Hz with each frequency being twice the preceeding frequency. The last frequency may be higher than 20000 Hz, but it will not be less.

If only f1 is specified, (not f1 or st), a single frequency analysis will be done. If f1 and f2 are specified but not st, the two frequencies f1 and f2 will be used. (Assume st to be 2) "S100,200" will do a two point analysis for 100 Hz and 200 Hz. "S200" will test 200 Hz only. "S0" or "S" (no numbers) will do a d.c analysis.

If f1 is not specified or zero, a dc analysis will be done. Logarithmic sweep is not allowed from d.c. A linear sweep will be done instead.

Using the "S" command, the voltage, dBV and phase at the highest numbered node will be displayed on the screen for each frequency. If the frequency is zero (d.c.) the phase and dB will not be printed, and the voltage will have a minus sign if it is negative.

If data for other nodes is desired, append ":N=n1,n2,..,nn" to the command. This will result in data for all specified nodes instead of the highest node. If data for only one node is needed, it is recommended that it be the highest numbered node since execution time is slightly faster.

If data for all nodes is needed, use the "A" (analyze) command. The format is the same.

If option "L" is inserted following the letter "S", (as in SLn1,n2) the results are printed on the printer.

If option "D" is specified, the results are saved in a disk or tape file. As with all files, the program will ask for a file name and a heading before opening the file.

Option "A" (as in "SAn1,n2") will cause the same nodes to be used for this analysis as for the previous S,W,O,C or I command in which nodes were specified.

Options "+" or "-" will cause all of the circuit values to be shifted according to the flags set by the most recent worst case analysis ("W" command). Option "P" will cause the worst case phase flags to be used instead of the worst case amplitude flags.

The options may be combined to perform the desired functions. For example, the command "SDL100,200,10:N=3,7" tells the computer to analyze the circuit from 100 Hz to 200 Hz, in 10 Hz steps, display the results for nodes 3 and 7, and print them on the line printer and save them as a disk file.

The process may be stopped at any time by pressing control-c. It may be necessary to hold it for a few seconds for it to be recognized, especially when analyzing large circuits. As before, breaking when data is output to the disk will cause the directory to not be updated, leaving a useless file.

You may pause execution by pressing control-s. Hitting any key will resume execution. It may take a few seconds for pause to be recognized.

Execution time will vary from less than one second per frequency to about one minute per frequency depending on the complexity of the circuit. It is rare that it will take more than 10 seconds per frequency. A d.c. analysis will take about one half as long as an a.c. analysis.

A detailed analysis may be done by the "A" command. This is the same as the "S" command except that data for all nodes is generated.

Interconnecting circuits, very large circuits (I command)

```
The I command is used for interconnecting circuits.
     I{opt}{:N=n1{,n2}{,...,nn}
Parameters:
     n    =  node numbers
Routing options:
     L    =  data to line printer
     D    =  data to disk file
     S    =  inhibit line printer
     X    =  inhibit disk
Analysis options:
     A    =  use previous ":N" optional nodes
     V    =  use previous worst case voltage flags
     P    =  use previous worst case phase flags
     +    =  use previous worst case increased flags
     -    =  use previous worst case decreased flags
```

Very large circuits can be analyzed by breaking them up to smaller circuits. The "I" command behaves similar to the "S" command except that the input data is taken from a disk file (generated by a previous "SD" command). "ID" will save the results in a disk file for even bigger circuits. This command can also be used to cascade identical stages.

Most options are handled the same as under "S" or "A".

After entering the command, the program will ask for the "old file name". Enter the file name to read. If the D option was specified, the program will also ask for a "new file name". This is the file for writing the new data. Be careful not to interchange the two file names, as it is possible to wipe out a valued file. If "new file name" already exists, the old file will be deleted and replaced by a new file of the same name.

The frequencies are taken directly from the file, so this additional analysis uses exactly the same frequencies. The resultant voltages are multiplied by the previous results. This has the same effect as changing all fixed sources to sources controlled by the output of a previous analysis. Normally, the file used as input should have data for only one node.

When the file is read, the input data (previous results) will be displayed on the screen. The results of the analysis will be displayed on the next line, so the display will show two lines for each frequency. The old data from the file is not sent to other devices such as disk or printer regardless of any options. Only the new results are sent to the printer or disk.

When connecting circuits together to make larger ones, be sure to take into account the loading that one circuit will place on the other. The program does not automatically do this. You can do it either by loading the first (in its analysis) with an impedance equal to the input impedance of the second block, or by making the source impedance in the second circuit equal to the output impedance of the first.

Comparisons (C command)

```
The "C" command compares the circuit in memory to a
previous circuit.
     C{opt}{:N=n1{,n2}{,...,nn}
Parameters:
     n    =  node numbers
Routing options:
     L    =  data to line printer
     D    =  data to disk file
     S    =  inhibit line printer
     X    =  inhibit disk
Analysis options:
     A    =  use previous ":N" optional nodes
     V    =  use previous worst case voltage flags
     P    =  use previous worst case phase flags
     +    =  use previous worst case increased flags
     -    =  use previous worst case decreased flags
```

The current circuit may be compared to another by the "C" command. It works similarly to the "I" command, except that the output is the voltage ratio, dB difference and phase difference between the two networks. It can also be used for comparison to a standard, such as RIAA equalization. To do this, a file is created (by an editor or a previous run) with the standard data, and is used for the comparison.

The output format is the same as the "I" command. The old file data is displayed as it is read, then the comparison result is displayed on the next line. Only the comparison result is sent to the printer or disk.

Worst case analysis (W command)

A worst case analysis is done by the "W" command.

```
     W{opt}f1{,f2{,st}}{:N=n1}
Parameters:
     f1   =  start frequency
     f2   =  end frequency
     st   =  step size if positive
             step multiplier if negative
     n1   =  node number   (default is highest node)
Routing options:
     L    =  data to line printer
     D    =  data to disk file
     S    =  inhibit line printer
     X    =  inhibit disk
Analysis options:
     A    =  use previous ":N" optional nodes
```

Worst case analysis may be done by the "W" command. Each branch is increased individually by its tolerance to determine whether the output and phase increases or decreases for an increase in its value.

After each branch that has a non-zero tolerance is tested, all branches are shifted for increased output, then decreased output, then leading phase, then lagging phase, and four sets of relative data are printed. The direction flags are stored for future use by other commands such as "S", "C" and "I".

If node numbers are specified in the command (by the "N" option), only one may be specified. If no node numbers are specified, the highest node will be used.

The display will consist of the reference results (absolute) followed by the relative result of varying each branch. The number under "volts" is a multiplier. 1 means no change, 1.1 means a 10 % increase, .9 means a 10 % decrease, etc. The numbers under dB and phase are differences resulting from the tolerance.

If a worst case analysis is done for d.c., the same rules apply except that dB and phase are not printed. A negative relative voltage means that the output voltage reverses polarity.

Any component whose effect is very small (within the computer's subtraction accuracy) will not be shifted for the final four worst case calculations.

It is recommended that all known non-critical components be given zero tolerance, because this will result in faster execution.

When used with other commands later, the "P" option will reference the phase flags instead of the voltage flags. The "V" option will cause the voltage flags to be used. The "+" option will vary all components for increased output or leading phase. The "-" option will vary all components for decreased output or lagging phase, as determined by the last worst case run. For example, "S+P100,500,100" will produce a sweep from 100 Hz to 500 Hz in 100 Hz steps, shifting all values according to the most recent worst case analysis for maximum leading phase. This could be used to show the shift in a bandpass filter center frequency.

It is permissible to sweep the "W" command, but only the flags from the last frequency will be saved for later use.

Dynamic, Temporary Modification (T command)

The circuit may be dynamically modified by the "T" command. This may be followed by a number which represents the number of steps to be used. For example, "T3" specifies 3 steps.

The sign of the number of steps represents the type of steps to be used by the "M" command. A positive number represents linear steps; a negative number represents logarithmic steps. "T" with no number following it results in one step. This command is the equivalent of putting variable components in the circuit.

The program will display a double prompt. At this prompt, the "A", "L", "M", "S" and "W" commands are allowed. All commands except "M" are delayed until receiving a "G" (go) command. The "D" option should not be used, since the program will ask for the file name (from the keyboard) again for each trial.

The "M" command should be used to modify the circuit. The format is the same as normal, except that it expects two new values, and cannot modify tolerance. The program will vary this branch from the first number to the second number in log or linear steps as specified by the "T" command. If only one number is entered, that value will be used for all steps. All values will be restored to their original values after the "T" command is finished. As many as 8 "M" commands may be executed here. The branches are varied together, not as nested loops. All "M" commands will be done first regardless of the actual order they are entered.

This would normally be followed by an analysis command such as "A" or "S", or an "L" command to list the modified circuit. The format is the same as normal. After this, the program will modify the circuit according to the "M" commands to each step, then analyze it according to the "A" or "S" command. This process will be repeated for the number of times specified in the "T" command. The program then returns to the normal prompt. All values are restored to their former values. The circuit list has not been changed.

This feature is intended for determining the effect of varying circuit parameters by controls or substitution, or for component optimization. It is not a worst case analysis. ("W" is a true worst case analysis, and can be done within the "T" loop.)

Batch mode (F command)

The commands may be entered from a disk file automatically by the "F" command. This is comparable to the CP/M submit command except that it is totally within this program.

To transfer control to the disk, enter the command "F". The program will then ask for the file name. Enter it in the standard format. The heading (file's first record) will be displayed and control will be transferred to the disk or cassette.

This is preferable to the dos facility because it is faster and can be aborted by pressing control-c without losing the circuit data in memory.

Control-s will cause execution to pause. Any key will cause it to continue.

Disk control files may be created by any ASCII format editor such as most word processors, or the CP/M editor.

The file must end with the "K" command or another "F" command. The "K" command returns command to the keyboard. An "F" command transfers control to another disk file. Its name must be on the next line. The name is actually the last record of the file. These commands are not nested. The new file replaces the old.

Any error will cause control to be returned to the keyboard.

Another mode of automatic operation can be done with the "T" command. A group of commands can be input by the keyboard and run as a group. Only listing and analysis commands can be run this way. File names cannot be entered this way. See "Dynamic modification" (page 20) for further information on this method.

Reset, select disk (D command)

The "D" command will reset the disk status, allowing a disk change, and will change the default disk. The command "D" followed by a letter will select a new drive to be used as the default drive for disk i/o. For example, "DC" selects drive "C" as the default drive. It also resets the disk status, allowing a disk change. The command "D" alone will reset the disk status and keep the previously selected default drive.

Quit (Q command)

The "Q" command ends this program and returns to the calling program, (usually dos or basic) and releases all memory it used. After this command, it will probably be necessary to reload the program and all data.

Pausing

Entering control-s at any time will cause execuition to pause. It may be necessary to hold it for a few seconds to be recognized. Hitting any key will cause execution to resume where it left off.

Breaking in

Any procedure in progress can be terminated by control-c. It may be necessary to hold it for a few seconds to be recognized. This will always return to the prompt, and return control to the keyboard. Options set by the "O" command are not affected.

Options (O command)

Options may be set by the "O" command.

```
     O{opt}{:N=n1{,n2}{,...,nn}
Parameters:
     n    =  node numbers  (default is highest node)
Routing options:
     L    =  all data to line printer
     D    =  all data to disk (spool) file
     S    =  cancel line printer
     X    =  cancel disk,  close file
Analysis options:
     A    =  use ":N" optional nodes
     V    =  select voltage as the worst case default
     P    =  select phase as the worst case default
```

Output may be routed by the "O" command, or by options in the individual commands. The "X" command cancels all output routing.

This routing applies only to data output by the computer. Requests, prompts and entries are not routed and will be displayed on the screen only. All output is displayed on the screen regardless of where else it is routed. The "O" command remains in effect until cancelled by another "O" command or an "X" command. It may be temporarily changed by options in each command.

There are two types of disk files this program can create. The most common is the local type, which are enabled by the individual operations. The other type is a global type which will route all output to a disk file. This is intended for use as a type of spool, that is used for printing later, or for further editing by a word processor. The "D" option in the "O" command creates this spool file. The "D" option in individual commands creates local files. Only the local type may be directly used for input later. It is possible to have both types at the same time as two different files. A global file, as created by the "O" command can be temporarily inhibited by the "X" option in an individual command.

File format

All disk files are in a standard ASCII format. They are compatible with most word processors and the CP/M editor. The file format is exactly the same as the display.

They are also compatible with the Microsoft Basic print# format. They can be read by the line input# command, but some elaborate string manipulations are necessary to extract the data due to the variable line legnth. The utility provided "BRANCH.BAS" is an example of how to extract this information.

All files have a header line at the beginning of the file which is displayed on the terminal and ignored. This line must be present.

Any record that begins with * will be displayed and ignored. Any text in any record that follows the required information is also ignored and may be used as comments.

Electronic Circuit analysis is extremely tolerant of variations in the file format. Generally, any format that works from the keyboard will also work from a file.

Component files (generated by the LD command, loaded by the BD command) contain 4, 5 or 6 entries per record. These are (in order): branch number (optional), component type, two node numbers, the value (resistance, etc.) of the component, and the component tolerance (optional). The component type is a letter, the value and tolerance are floating point, and the others are positive integers. Entries must be separated by commas or spaces and have a carraige return at the end of each branch. The file must be terminated by a null line or control-z. The output format is identical to that displayed on the screen.

Results files contain 3 or 5 entries per record. These are (in order): frequency, node number, voltage, dB, phase. All except node number are floating point. Entries can, as above, be either separated by commas or spaces and must have a return at the end. This file also has a header line which must be present. The computer ignores node numbers and dB gain when reading the file. These are present in the output so the file can be merged with a text file for use in a report, and for plotting and branch data utilities. The output format is identical to the data displayed on the screen. The file must be terminated by a null line or control-z.

Control files are simply a list of entries of the type that would be input from the keyboard, in the same format. Anything that can be input from the keyboard can also be done from a control file. The data must be in the exact form you would enter from the keyboard. This file, like the others, must have a header line. Electronic Circuit Analysis cannot create its own control files, they must be created by another method such as any ASCII format editor. This file must be terminated by the "K" or "F" command. The "F" command must be followed by the file name to be used.

Branch current, power

A utility (BRANCH.BAS) is provided to generate the voltage, current, and power for all branches. It is written in Microsoft Basic.

To use it, run "BRANCH" from Basic. It is self prompting and will ask for all the information it needs. It requires two files, generated by ECA. The first is the component list, generated by "LD". The second is the data file, generated by "AD".

The program will first ask if a printout is needed: Respond with Y or N. It will then ask for the component file, and then the data file. Respond with the full name of the file. The program will display the data file, then it will display the calculated data for all branches. The output will be in groups by frequency. If output is not routed to the printer, the program will pause at each frequency and display "?". Enter or return will cause the program to continue to the next frequency.

Accuracy

In normal use, this program should be as accurate as any instruments you may have to measure the finished circuit with.

There are several possible causes of inaccuracy. Most of these can be controlled in the entry of the circuit.

The most common cause of inaccuracy is leaving out parts. All parts should be included. Frequently, strays are omitted, since they don't show on the schematic. They are usually more significant than you might think. The 10 pf input capacitance of a common op-amp was enough to cause the analysis of an audio filter to be 3 dB out of spec. R.F. circuits will have strays at just about every point. Also, inductors always need a resistor to set the Q, since no real inductor has infinite Q.

Another cause of inaccuracy is improper modeling of the active parts. An example of this is using a single pole model of an op-amp when a two pole model is needed. Models of many active devices are provided in the "Models" section of this manual. They are provided as a guide, and they are approximations. A particular application may require a different or more complete model than what is provided. A good text in electronic theory may help to model the device properly.

A common variation on the improper models problem is that the data published by device manufacturers is often inaccurate. It is always incomplete. The data provided on low frequency power transistors, for example, seldom includes information on its capacitances. This data is absolutely necessary to properly analyze any feedback amplifier. Perhaps when this type of detailed analysis program becomes common, the manufacturers will provide more complete data on their devices.

Another possible problem is the accuracy of the program itself. This program uses 4 byte floating point for all its internal calculations, for 6 digit accuracy. This may not be good enough if admittances connecting to the same node vary widely. In general, an admittance should be within a range of 100,000 times any other admittance connecting to the same node. Usually, if this problem exists, the results will change if the nodes are renumbered. The errors caused by this are often very large. A complete analysis (A command) will usually show where the problem is. A future release of this program will have an increased precision option to solve this problem.

In a d.c. analysis, a resistor of the value entered is substituted for an inductor. This substitutes an arbitrary low value resistor. Occasionally this will cause errors. The branch could not simply be shorted because the method of analysis will not allow a direct short between branches. Normally, this resistor substitution will be much lower in value than any resistor already in series with the inductance that it will not have any appreciable effect on the result.

Illogical numbering of the nodes also decreases accuracy. Generally the nodes should be numbered sequentially from input to output. If a very few nodes have many connections, these should have high numbers. A circuit having sidechains should have the sidechains numbered first and the main path last. This also improves execution time substantially, and is worth doing on a large circuit for this alone.

A circuit that is unstable at the test frequency will produce very wrong looking results. Usually the phase will be reversed from what makes sense. A sweep past the critical frequency will often cause a sharp phase shift or a near infinite voltage. A complete analysis (A command) will be helpful in finding this.

Error Messages

Most error messages will be clear to the user when they are presented. All errors will terminate the command in progress and return to the prompt.

"Overflow" -- Usually occurs when an integer entry from the keyboard exceeds 255. It may also if a floating point keyboard entry exceeds 1E38. Also may occur if an internal calculation during analysis exceeds 1E38. If this occurs, check the component value list for improper values.

"Illegal math function" -- usually means division by zero -- Usually occurs on "C" or "I" commands when the file being read is not of the proper format or if the voltage specified is zero. Also may occur on analysis if sources are entered incorrectly. All sources must be entered as two consecutive branch statements.

"node x unconnected .. no solution" -- Occurs on analysis. Usually it means exactly what it says, but sometimes it is a floating sub-circuit with connections only within itself. Another explanation could be "voltage undefined at node x". Strange behavior of the circuit such as oscillation or exact cancellation of admittances could cause this error message.

In case of difficulty

If you have any problems running this program, we do want to hear from you. As time passes there will be enhancements to the program, and there could be errors not presently known.

Considerable effort was expended to test the program on several different systems. It is possible that an incompatibility will still exist. We would like to know af any incompatibility that arises, and will attempt to make it compatible.

Registered users will be notified of any enhancements or significant changes made to the program. Repairs of problems will be provided to registered users at no charge. Enhancements will be provided at a nominal charge to cover shipping, handling and any new materials. It is our intention to release an enhanced version of this program in the future. Any comments you may have regarding improvements would be appreciated.

If you have problems, they should be reported to:
Tatum Labs
P.O. Box 722
Hawleyville, CT 06440
Phone: 203-426-2184

Telephone answering personnel are not technically knowledgeable of this program, so it is recommended that you write rather than phone. It would be helpful, and sometimes essential, to send a disk demonstrating the problem.

Notes

All node numbers up to the highest number used must be used. Do not leave any blank nodes.

Nodes may be added or removed by the "B" command, provided that all nodes up to the highest number used are connected.

If you must delete a node, tie it to ground with a 1 ohm resistor.

Component values must be finite and not zero.

Negative values are permitted.

Branches do not have to be consecutively numbered. Blank branches are permitted.

Although nodes can be numbered in any sequence, execution time will be faster if the nodes with the most connections have the highest numbers. Improvements of as much as 4 to 1 have been obtained with this technique. This is very significant with large circuits. Connections to the highest numbered nodes require much fewer operations than those to lower numbered nodes.

It is possible that the worst case analysis may not find the true worst case due to interactions. In some circuits, shifting one value will cause shifting another value to have the opposite effect from what it had when shifted individually. This effect, if present, is usually very small and insignificant. It can appear large if, for example, a filter is shifted far from resonance.

Command summary

A analyze A{opt}f1{,f2{,st}}
Complete analysis for all nodes.
Options supported : L, S, D, X, P, V, +, -
f1 = start frequency
f2 = end frequency
st = step size if positive
step multiplier if negative

B build B{D}{b1{,b2}}
Enter new circuit parts
D = load from disk
b1 = start branch
b2 = end branch

C compare C{opt}{:N=n1{,n2}{,...,nn}}
Compare results to previous analysis
Options supported : L, S, D, X, P, V, +, -, A
":N=" node numbers

D reset disk D{dr}
Resets disk tables, to allow changing disks
dr = new default disk

F file F
Transfers control of program to disk or cassette.
"K" command ends file

G go G
Executes the sequence of commands just entered under "T"

I input I{opt}{:N=n1{,n2}{,...,nn}}
Analyze using previous data as input
Options supported : L, S, D, X, P, V, +, -, A
":N=" node numbers

K keyboard K
Returns control of program to keyboard

L list L{opt}{b1{,b2}}
Prints component list.
Options supported : L, S, D, X
b1 = start branch
b2 = end branch

M modify M{T}b1{,b2}
Modifies single branch or group of branches.
T = modify tolerance (else modify value)
b1 = start branch
b2 = end branch

O options O{opt}{:N=n1{,n2}{,...,nn}}
Set default options
Options set : L, S, D, X, P, V, A
":N=" node numbers if A option set

Q quit Q
Terminates program.
Releases all memory used.

R reset R
Clears component list

S sweep S{opt}f1{,f2{,st}}{:N=n1{,n2}{,...,nn}}
Simple analysis
Options supported : L, S, D, X, P, V, +, -, A
f1 = start frequency
f2 = end frequency
st = step size if positive
step multiplier if negative
":N=" node numbers

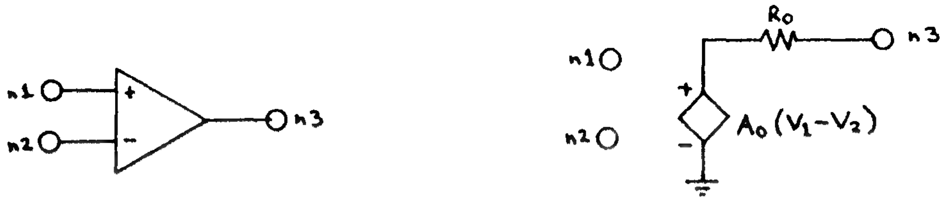
T tweek Tst
Dynamically varies circuit and retests
st = number of steps
linear if positive, log if negative

W worst case W{opt}f1{,f2{,st}}{:N=n1}
Worst case analysis
Options supported : L, S, D, X, A
f1 = start frequency
f2 = end frequency
st = step size if positive
step multiplier if negative
":N=" node number (only one)

X cancel routing X
Cancels all options.
Cancels anything set by "O"

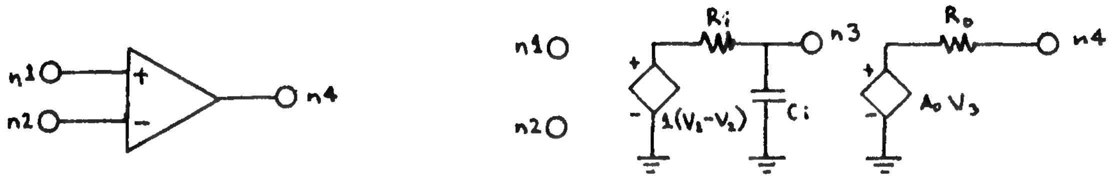
Models:

Op-amp, infinite bandwidth



3 nodes, 2 branches:
V,n1,n2,open loop gain
R,n3,0,open loop output resistance

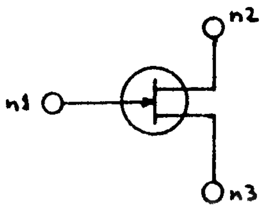
Op-amp, single pole

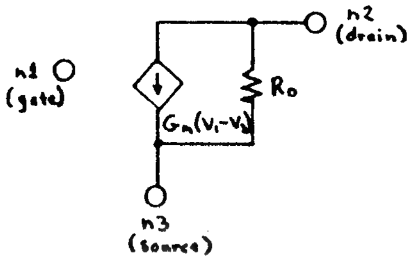


4 nodes, 5 branches:
V,n1,n2,1
R,n3,0,Ri
C,n3,0,Ci
V,n3,0,open loop gain
R,n4,0,open loop output resistance

Select Ri, Ci for dominant pole of open loop response.
One internal node is added (n3).

Fet, infinite bandwidth

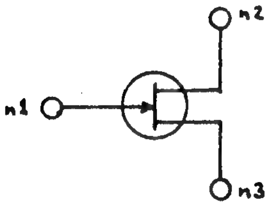


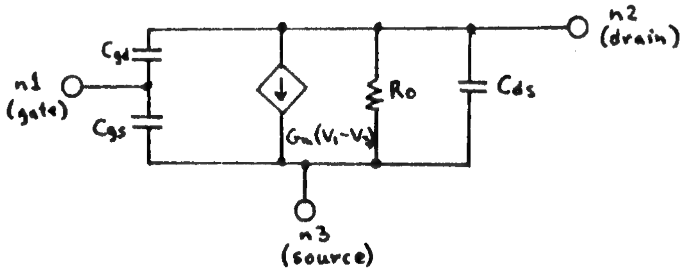


3 nodes, 2 branches:
I,n3,n1,transconductance
R,n2,n3,Ro

Ro is the output resistance. It can be read from the characteristic curves as the slope of the flat portion (dv/di). Often it is not specified, or its reciprocal (yos) is specified.
All fets are modeled the same regardless of polarity, type, or use in circuit.
Transconductance may be given as gm or yfs.

Fet, finite bandwidth





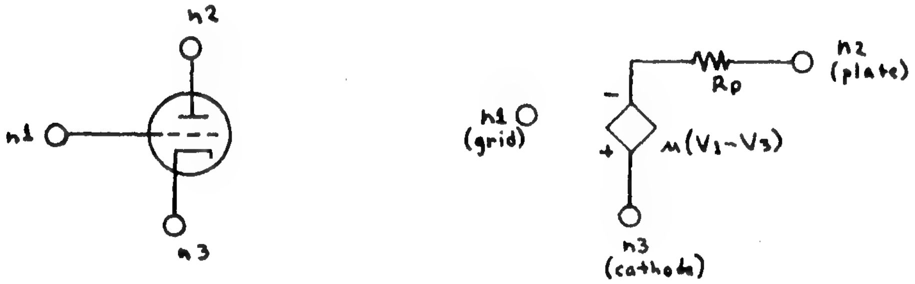
3 nodes, 5 branches:
I,n3,n1,transconductance
R,n2,n3,Ro
C,n1,n3,Cgs
C,n1,n2,Cgd
C,n2,n3,Cds

Cgs, Cgd and Cds should be available from the data sheet on the device being used.

Vaccuum tube

Same as fet, except substitute grid for gate, cathode for source, plate for drain.
Add capacitances to screen and suppressor grids, if they are present.
If screen or suppressor grids are used as inputs and are not connected to other elements, a different, more complex model is necessary. This is usually done only in nonlinear circuits which can not be fully analyzed with this program.

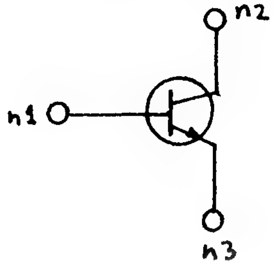
Vaccuum tube (triode) voltage source model

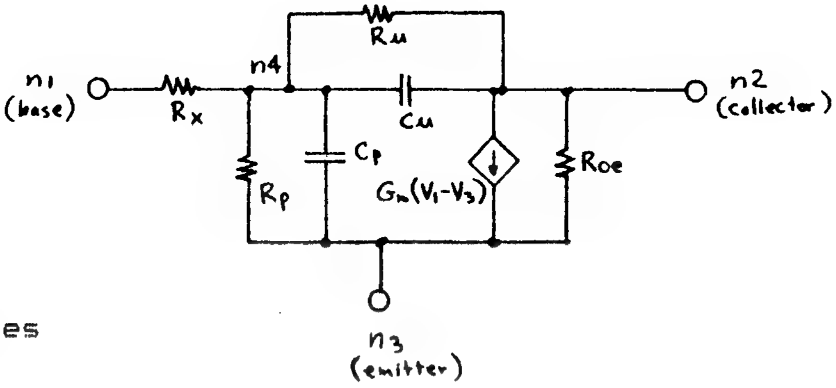


3 nodes, 2 branches:
V,n3,n1,amplification factor
R,n2,n3,plate resistance

V branch replaces I branch.
To add capacitances, see fet finite bandwidth.

Transistor, a.c. high frequency
Hybrid - pi model

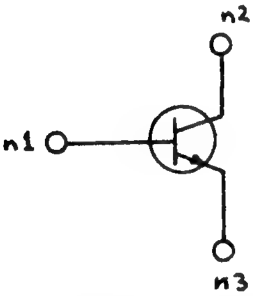


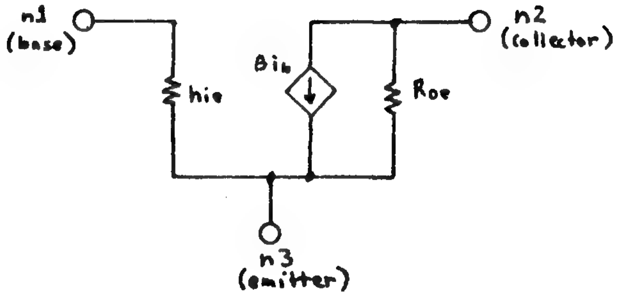


4 nodes, 7 branches
I,n3,n1,Gm
R,n2,n3,Roe
R,n4,n2,Ru
C,n4,n2,Cu
R,n4,n3,Rp
C,n4,n3,Cp
R,n1,n3,Rx

One internal node is added (n4).

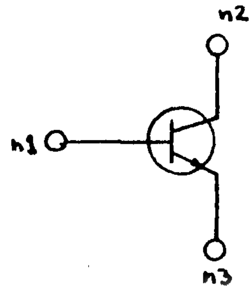
Transistor, a.c. low frequency

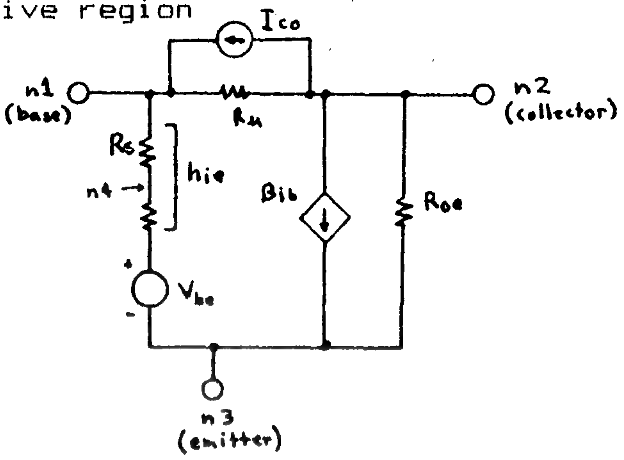




3 nodes, 3 branches
I,n3,n1,beta/hie
R,n2,n3,Roe
R,n3,n1,hie

Transistor, d.c. active region

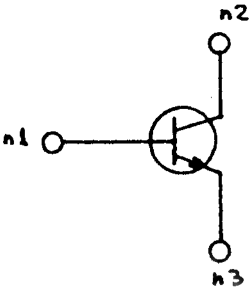


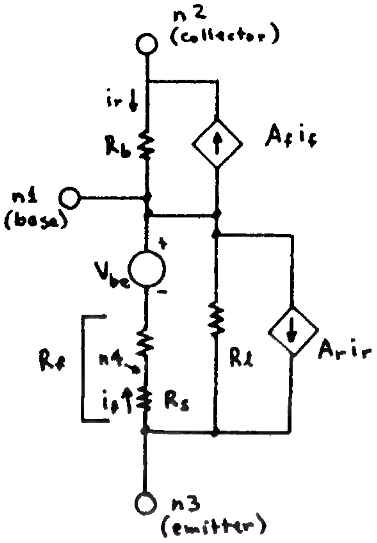


4 nodes, 7 branches
I,n4,n1,beta/Rs
R,n2,n3,Roe
I,0,0,Ico
R,n2,n1,Ru
V,0,0,Vbe
R,n4,n3,hie-Rs
R,n1,n4,Rs

Rs is a sense resistor for sensing base current.
  The best value is about 1/2 of hie
Vbe is usually 0.6 V for silicon or 0.2 V for germanium
Ico and Ru represent collector to base leakage
One internal node is added (n4)
This model is for NPN transistors.
  For PNP, reverse Vbe, Ico

Transistor, d.c. overdriven

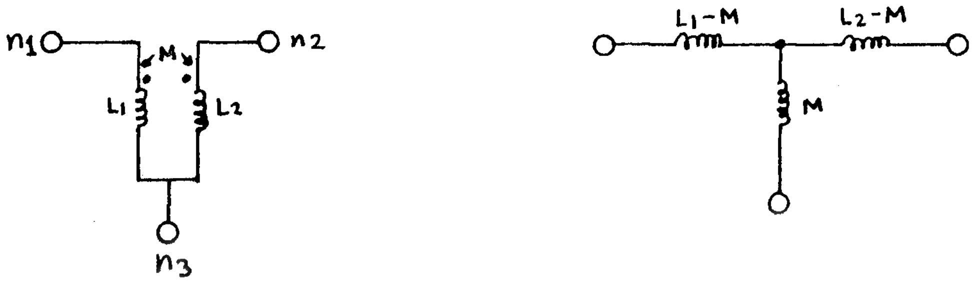




```
4 nodes, 6 branches
I,n3,n4,Af/Rs
R,n1,n2,Rb
I,n2,n1,Ar/Rb
R,n1,n3,Rl
V,0,0,Vbe
R,n4,n1,Rf-Rs
```

Rs is a sense resistor for sensing emitter current.
  The best value is about 1/2 of Rf
Rl is an arbitrary high value to satisfy this program's
  requirements for a direct shunt on a current source
This model is for an NPN transistor. For PNP, reverse Vbe.
Typical values: Rf = 20, Rb = 1E6, Af = 0.98, Ar = 0.90
                Vbe = 0.6 (silicon) or 0.2 (germanium)
One internal node is added (n4).

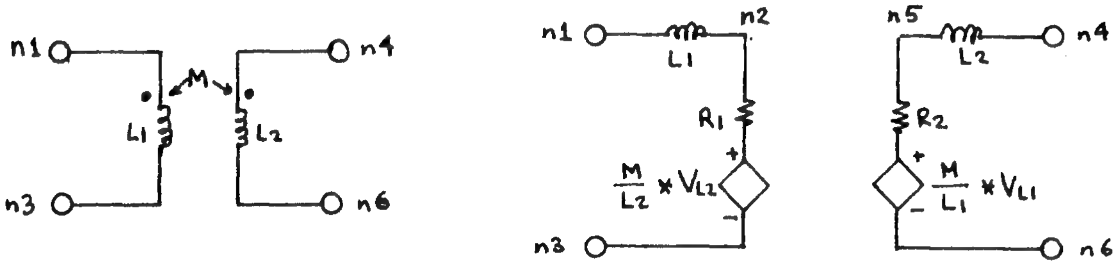
Mutual Inductance -- T equivalent
(common node between ports)



```
4 nodes, 3 branches
L,n1,n4,L1-M
L,n2,n4,L2-M
L,n4,n3,M
```

One internal node is added (n4)

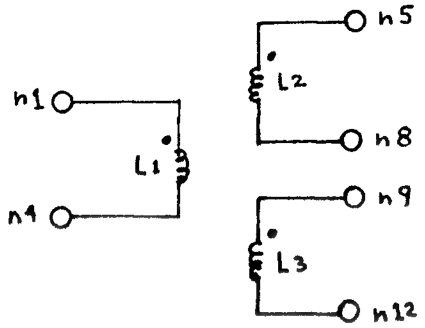
Mutual inductance, two inductor general case

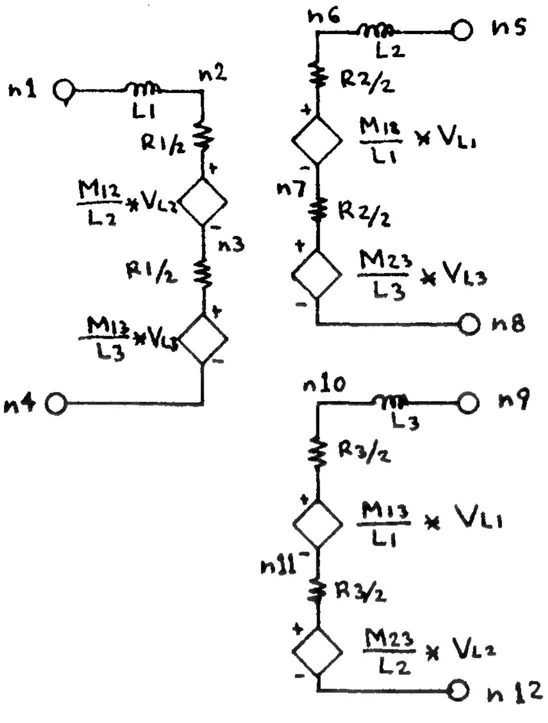


```
6 nodes, 6 branches
L,n1,n2,L1          primary
V,n4,n5,M/L2
R,n2,n3,R1
L,n4,n5,L2          secondary
V,n1,n2,M/L1
R,n5,n6,R2
```

Two internal nodes are added (n2,n5)
This equivalent circuit actually adds only one branch over the T equivalent, since the resistors are necessary in a complete simulation even with the T equivalent.

Mutual inductance, more than two inductors





12 nodes, 15 branches

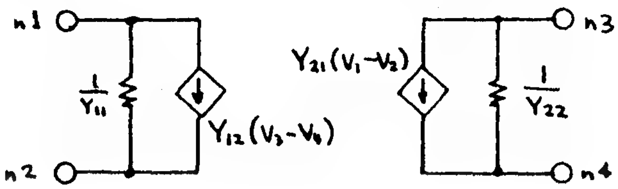
```
L,n1,n2,L1
V,n5,n6,M12/L2
R,n2,n3,R1/2
V,n9,n10,M13/L3
R,n3,n4,R1/2

L,n5,n6,L2
V,n1,n2,M12/L1
R,n6,n7,R2/2
V,n9,n10,M23/L3
R,n7,n8,R2/2

L,n9,n10,L3
V,n1,n2,M13/L1
R,n10,n11,R3/2
V,n5,n6,M23/L2
R,n11,n12,R3/2
```

6 internal nodes are added
Controlled sources are connected in series
The series resistance of the inductors is split between the sources

General case, Y parameters



```
I,n1,n2,Y21
R,n3,n4,1/Y22
I,n3,n4,Y12
R,n1,n2,1/Y11
```

Only real parameters can be modeled this way.
Imaginary parameters must be modeled by adding capacitors and inductors.

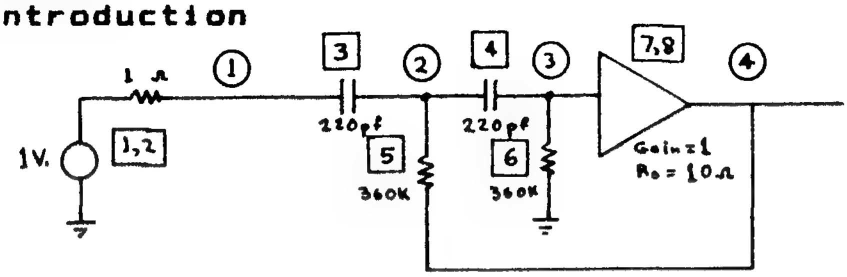
Parameter Transformations

	y11	y12	y21	y22
z	z22/|z|	-z12/|z|	-z21/|z|	z11/|z|
h	1/h11	-h12/h11	h21/h11	|h|/h11
g	|g|/g22	g12/g22	-g21/g22	1/g22
a	a22/a12	-|a|/a12	-1/a12	a11/a12
b	b11/b12	-1/b12	-|b|/b12	b22/b12

|z| = determinant of z , etc. (z11*z22-z12*z21)

Examples

Example 1
One amp high pass filter
Introduction



To analyze this circuit:

- Number the nodes. (Use 0 for ground. Number the other nodes consecutively.
- Number the branches (Any numbers will do)
- Enter circuit values:

```
-->  B           The "B" command builds a circuit
 1 V,0,0,1          A fixed voltage source -- 1 volt
 2 R,1,0,1          Resistor, 10 ohms, in series with the
                      voltage source the combination
                      connects between node 1 and ground
                      the "in phase" node is node 1
 3 C,1,2,220E-12    Capacitor, 220 pf between nodes 1 and 2
 4 C,2,3,220E-12    Capacitor, 220 pf between nodes 2 and 3
 5 R,2,4,360E3      Resistor, 360 K between nodes 2 and 4
 6 R,3,0,360E3      Resistor, 360 K between node 3 and ground
 7 V,3,0,1          Controlled voltage source (amplifier).
                      Gain = 1 Input terminals: 3 and ground.
 8 R,4,0,10         Resistor, 10 ohms, in series with source
                      Amp output terminals are 4 and ground.
                      Node 4 is in phase with node 3.
 9                  Hit enter or break to end this mode
-->                 What now?
```

*Now we have entered the circuit. It is a good idea to list it to see it you entered it correctly:

```
  --> L                              "L" means list.
  branch   type   nodes   value   tolerance
      1      V     0  0    1            Notice that the display is
      2      R     1  0    1            formatted and that some
      3      C     1  2    2.2E-10      numbers seem to have changed
      4      C     2  3    2.2E-10      they are really the same.
      5      R     2  4    360000       Also, notice the blank
      6      R     3  0    360000       column marked "tolerance".
      7      V     3  0    1            We will take care of that
      8      R     4  0    10           later.

  -->
```

* If it looks right, save it on a disk or tape.

```
  --> LD                                     "LD" means list to disk
  New file name?  EX1.CKT                        Enter the file name
  Heading?
  Type up to one line, so you know what this file is.
  branch   type   nodes   value   tolerance
  etc.                                 The values will be displayed
                                           on the screen,
  -->                                      just as in "L" above.
```

* Now that we are covered in case of disaster, let's try to analyze it. It is some kind of audio filter, a high pass, but I don't know where the corner is. Probably the best thing to do is sweep the whole audio range. A log sweep from 31.25 Hz to 16 kHz in octave steps works nice. A negative step size means a log sweep. "S 31.25,16000,-2" means sweep from 31.25 Hz to 16000 Hz and multiply each frequency by 2 to get the next step.

```
-->  S 31.25,16000,-2
          freq          node    volts           dB     phase
         31.25           4    2.41779E-04    -72.3  +178.1160
         62.5            4    9.66415E-04    -60.3  +176.3860
         125             4    3.85447E-03    -48.3  +172.8550
         250             4    .0152417       -36.3  +165.8040
         500             4    .0583006       -24.7  +152.0480
         1000            4    .198485        -14.0  +127.0830
         2000            4    .497623         -6.1   +90.2699
         4000            4    .79847          -2.0   +53.3472
         8000            4    .940645         -0.5   +28.2005
         16000           4    .984469         -0.1   +14.3170
-->
```

* What I really wanted was for the 6 dB down point to be 1 kHz. I think doubling both 220 pf capacitors will accomplish this.
* The computer displays the data for the branch as it now stands, and waits for an input. Type in the new value. The computer will display the branch as it is after modification.

```
-->  M3                     Modify branch 3
    3     C     1   2       2.2E-10         440E-12
    3     C     1   2       4.4E-10
-->  M4                     Modify branch 4
    4     C     2   3       2.2E-10         440E-12
    4     C     2   3       4.4E-10
-->
```

* Let's see if that worked.

```
-->  S1000      Sweep (sort of) 1000 Hz only
          freq          node    volts           dB     phase
        1000             4    .497623         -6.1   +90.2699
-->
```

* That looks ok, now sweep the whole range

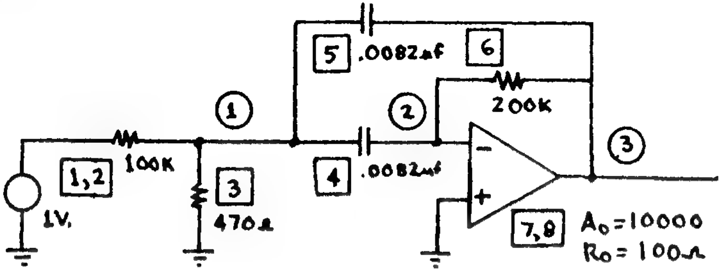
```
-->  S 31.25,16000,-2
        freq        node    volts          dB    phase
        31.25        4    9.66415E-04   -60.3  +176.3860
        62.5         4    3.85447E-03   -48.3  +172.8550
        125          4    .0152417      -36.3  +165.8040
        250          4    .0583006      -24.7  +152.0480
        500          4    .198485       -14.0  +127.0830
        1000         4    .497623        -6.1   +90.2699
        2000         4    .79847         -2.0   +53.3472
        4000         4    .940645        -0.5   +28.2005
        8000         4    .984469        -0.1   +14.3170
        16000        4    .99607         -0.0    +7.1866
-->
```

* I would like to know the 3 dB down point. Sweep linearly from 1000 Hz to 2000Hz in 100 Hz steps. A positive step size means to sweep linearly.

```
-->  S 1000,2000,100
        freq        node    volts          dB    phase
        1000         4    .497623        -6.1  +90.2699
        1100         4    .545153        -5.3  +84.8164
        1200         4    .58786         -4.6  +79.8771
        1300         4    .626027        -4.1  +75.3988
        1400         4    .660029        -3.6  +71.3315
        1500         4    .690276        -3.2  +67.6302
        1600         4    .717173        -2.9  +64.2544
        1700         4    .741106        -2.6  +61.1681
        1800         4    .762429        -2.4  +58.3395
        1900         4    .781456        -2.1  +55.7407
        2000         4    .79847         -2.0  +53.3472
-->
```

* That is good enough. Be sure to save the file. (We did, just after typing it in.) You will need it later.

Example 2
Bandpass filter
Worst case analysis



* This circuit is on your disk as "EX2.CKT"
* Instead of typing it in, we will just load it from the disk.
* "BD" builds a circuit from disk
* When it asks for a file name, type it in.

```
-->  BD
Old file name?  EX2.CKT
Example 2  --  bandpass filter
     1     V     0  0    1
     2     R     1  0    100000        1.0
     3     R     1  0    470           1.0
     4     C     1  2    8.2E-09       1.0
     5     C     1  3    8.2E-09       1.0
     6     R     2  3    200000        1.0
     7     V     0  2    10000        25.0
     8     R     3  0    100          25.0

-->
```

* The computer has read in the file. Notice two things that are different:

There is a title. This is just the title you typed in when you saved the file.

There is an extra column. This is the tolerance. It is entered when you enter the circuit just after the value. It is not displayed if it is zero.

* It is supposed to be a 2 kHz bandpass filter with a Q of 10. A linear sweep from 1000 to 3000 Hz in 100 Hz steps is a good starting point.

```
-->  S1000,3000,100
         freq        node    volts          dB    phase
         1000          3    .0639889      -23.9   -93.7478
         1100          3    .0755721      -22.4   -94.4275
         1200          3    .0896639      -20.9   -95.2552
         1300          3    .107333       -19.4   -96.2946
         1400          3    .130315       -17.7   -97.6498
         1500          3    .161627       -15.8   -99.5032
         1600          3    .206969       -13.7  -102.2060
         1700          3    .278345       -11.1  -106.5190
         1800          3    .40453         -7.9  -114.4080
         1900          3    .656753        -3.7  -132.1340
         2000          3    .976801        -0.2  -176.1970
         2100          3    .719885        -2.9  +137.3380
         2200          3    .462486        -6.7  +118.1920
         2300          3    .332261        -9.6  +109.8400
         2400          3    .259082       -11.7  +105.3460
         2500          3    .212961       -13.4  +102.5640
         2600          3    .181384       -14.8  +100.6780
         2700          3    .158435       -16.0   +99.3137
         2800          3    .141          -17.0   +98.2811
         2900          3    .127295       -17.9   +97.4714
         3000          3    .11623        -18.7   +96.8186
-->
```

* It apparently does have a center frequency of 2000 Hz and a bandwidth of 200 Hz, so the claims were correct. Before going into production on this, let's see how the performance will vary with component tolerances. "W2000" wil do a worst case analysis at 2000 Hz.

```
-->  W2000
branch   freq        node    volts          dB    phase
         2000          3    .976801        -0.2  -176.1970
 2       2000          3    .990131        -0.1    -0.0268
 3       2000          3    1.00185        +0.0    -5.7279
 4       2000          3    1.0065         +0.1    -5.7641
 5       2000          3    .996739        -0.0    -5.7645
 6       2000          3    1.01142        +0.1    -5.7736
 7       2000          3    1.00421        +0.0    +0.0184
 8       2000          3    .999999        -0.0    -0.0031
         2000          3    1.02775        +0.2   -11.4060
         2000          3    .933078        -0.6   +11.2060
         2000          3    .908923        -0.8   +21.6287
         2000          3    .943894        -0.5   -22.2835
branch   freq        node    volts          dB    phase
-->
```

* That is a lot of data. It needs an explanation. The first line is a reference. It is the same as a normal single point analysis at 2000 Hz. The next group of lines is relative outputs varying each branch individually. Branch 1 is not shown because it has zero tolerance. It says that when the value of branch 2 is increased by the tolerance, (1 %) the level drops 0.1 dB and the phase is not affected. Branch 3 causes a slight (less than 0.1 dB) level increase and the phase to lag the reference by 5.7 degrees. This is repeated for all branches. The four unlabeled lines are worst case conditions. The first is all branches shifted for increased level. The second is decreased level. The third is leading phase. The fourth is lagging phase.

* It is apparent that branch 2 affects the results very little. Let's try a 5 % part instead of the 1 % part that was used. "MT" modifies the tolerance.

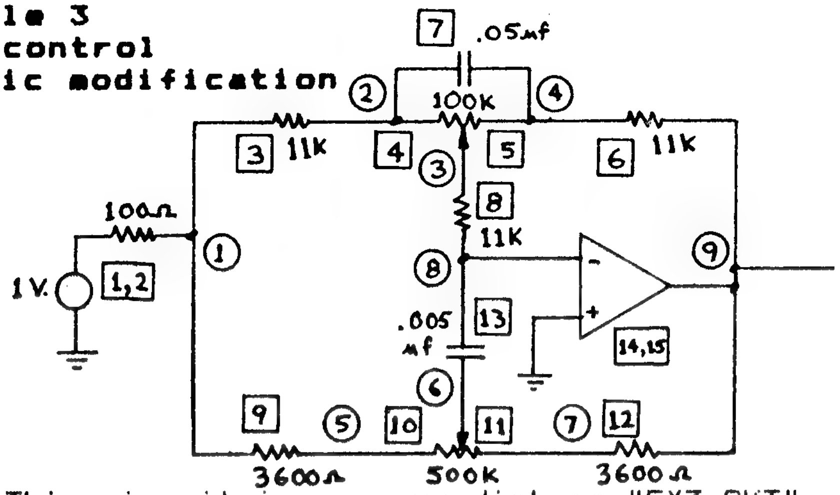
```
-->  MT2
    2    R    1  0   100000       1.0       5
    2    R    1  0   100000       5.0
-->  W2000
branch    freq        node   volts        dB    phase
          2000         3    .976801      -0.2 -176.1970
 2        2000         3    .952525      -0.4   -0.1291
 3        2000         3    1.00185      +0.0   -5.7279
 4        2000         3    1.0065       +0.1   -5.7641
 5        2000         3    .996739      -0.0   -5.7645
 6        2000         3    1.01142      +0.1   -5.7736
 7        2000         3    1.00421      +0.0   +0.0184
 8        2000         3    .999998      -0.0   -0.0031
          2000         3    1.0713       +0.6  -11.2918
          2000         3    .897934      -0.9  +11.1109
          2000         3    .946437      -0.5  +21.7243
          2000         3    .907459      -0.8  -22.3745
branch    freq        node   volts        dB    phase
-->
```

* It is still not bad, we will use a 5 % resistor for branch 2.
* What I really want to know is how the center frequency varies with component tolerance. Sweeping with the parts all shifted for worst case phase will give a good indication, mush better than for worst case amplitude.
* The "+" and "-" options of the "S" command request that worst case values are used. The "P" option requests that the phase data are used instead of the voltage data.

```
-->  S+P2000,2050,10
          freq        node    volts          dB    phase
          2000          3    .924481        -0.7 -154.4730
          2010          3    .958901        -0.4 -159.3870
          2020          3    .987762        -0.1 -164.6120
          2030          3    1.00918        +0.1 -170.0840
          2040          3    1.02162        +0.2 -175.7170
          2050          3    1.02418        +0.2 +178.5950
-->  S-P1940,2000,10
          freq        node    volts          dB    phase
          1940          3    .901323        -0.9 -164.5530
          1950          3    .921418        -0.7 -170.1870
          1960          3    .932808        -0.6 -175.9880
          1970          3    .934617        -0.6 +178.1580
          1980          3    .926823        -0.7 +172.3710
          1990          3    .910258        -0.8 +166.7640
          2000          3    .886405        -1.0 +161.4280
-->
```

* The center frequency (where the phase is exactly 180 degrees, in this case) has shifted to about 1965 Hz or 2025 Hz. Notice that the sweep is absolute, not relative as was the worst case test. This is really the worst case. In production it would not be this bad. Sweeping for worst case amplitude would not show as much shift.

Example 3
Tone control
Dynamic modification



* This circuit is on your disk as "EX3.CKT". Load it.

```
-->  BD
Old file name?  EX3.CKT
Example 3   tone control
    1     V     0  0   1
    2     R     1  0   100
    3     R     1  2   11000
    4     R     2  3   50000
    5     R     3  4   50000
    6     R     4  9   11000
    7     C     2  4   5E-08
    8     R     3  8   11000
    9     R     1  5   3600
    10    R     5  6   250000
    11    R     6  7   250000
    12    R     7  9   3600
    13    C     6  8   5E-09
    14    V     0  8   10000
    15    R     9  0   100

-->
```

* A quick check reveals that the frequency response is "flat", since all controls are centered.

```
-->  S100,10000,-10
        freq         node   volts          dB    phase
       100            9    .995968        -0.0  +179.8180
       1000           9    .990625        -0.1  +179.9280
       10000          9    .990408        -0.1  +179.9930
```

* We will use the "T" command to find the effect of one of the controls. We will adjust the control in 5 steps and make a frequency response run for each step.
* "T5" instructs the computer to sweep some parts linearly in 5 steps, according to the commands that follow. The commands that follow are "M" commands to tell the computer what to change, then "L" and "S" commands to tell the computer what to do with the modified circuit. The double prompt says that the computer is waiting for all the data before it does anything.

```
-->  T5
-->-->  M4
    4    R    2  3   50000          1,100000
-->-->  M5
    5    R    3  4   50000          100000,1
-->-->
```

* What we have just done is to modify branches 4 and 5 temporarily, in 5 steps. Branch 4 will vary from 1 ohm to 100 K. Branch 5 will vary from 100 K to 1 ohm. Note that the order is reversed. One will increase as the other decreases. Since it is a linear sweep, the total will always be 100001 ohms. Note that the computer seems to just ignore this entry. It really stores it for later use. We still have the double prompt. The next step is to do something with it. We will sweep the frequency in the same 3 steps for each set of conditions for branches 4 and 5. We will list branches 4 and 5 before each sweep so we know what we have.

```
-->-->  L4,5
-->-->  S100,10000,-10
-->-->
```

* The computer still hasn't done anything. It just stores the instruction and asks for more. Typing "G" will make it go and do the whole thing.

```
-->-->  G
branch  type  nodes   value  tolerance
    4    R    2  3   1
    5    R    3  4   100000
          freq        node   volts        dB    phase
         100           9   2.89719      +9.2 +124.9270
         1000          9   1.02862      +0.2 +165.7230
         10000         9   .990795      -0.1 +178.5330
branch  type  nodes   value  tolerance
    4    R    2  3   25000.8
    5    R    3  4   75000.3
          freq        node   volts        dB    phase
         100           9   1.72458      +4.7 +156.0110
         1000          9   1.00437      +0.0 +173.5540
         10000         9   .990545      -0.1 +179.3470
branch  type  nodes   value  tolerance
    4    R    2  3   50000.5
    5    R    3  4   50000.5
          freq        node   volts        dB    phase
         100           9   .995969      -0.0 +179.8180
         1000          9   .990626      -0.1 +179.9280
         10000         9   .990408      -0.1 +179.9930
branch  type  nodes   value  tolerance
    4    R    2  3   75000.3
    5    R    3  4   25000.8
          freq        node   volts        dB    phase
         100           9   .574606      -4.8 -156.3320
         1000          9   .977129      -0.2 -173.6970
         10000         9   .990272      -0.1 -179.3620
branch  type  nodes   value  tolerance
    4    R    2  3   100000
    5    R    3  4   1
          freq        node   volts        dB    phase
         100           9   .341256      -9.3 -125.0320
         1000          9   .954301      -0.4 -165.8600
         10000         9   .990272      -0.1 -179.3620
-->
```

* Everything has returned to normal. You could have uesd more steps if you wanted. You could even have asked for worst case for each step. Although this example shows a family`of curves, this method can also be used to optimize parts, by sweeping and noticing what happens to the output.

Example 4
Advanced file operations

* This example uses the same circuit as example 2, to demonstrate circuit comparison and connecting small circuits together to make one larger one.
* Part 1 -- what happens if we put the signal through the circuit twice?
* First, load the circuit file as you did in example 2. Then sweep it saving the results.

```
--> BD
Old file name?  EX2.CKT
Example 2  --  bandpass filter  parts list
    1    V    0  0   1
    2    R    1  0   100000       1.0
    3    R    1  0   470          1.0
    4    C    1  2   8.2E-09      1.0
    5    C    1  3   8.2E-09      1.0
    6    R    2  3   200000       1.0
    7    V    0  2   10000       25.0
    8    R    3  0   100         25.0

--> SD1800,2200,100
New file name?  EX4
Example 4 reference, (ex2 analyzed)
         freq        node   volts        dB    phase
        1800          3   .40453       -7.9 -114.4080
        1900          3   .656753      -3.7 -132.1340
        2000          3   .976801      -0.2 -176.1970
        2100          3   .719885      -2.9 +137.3380
        2200          3   .462486      -6.7 +118.1920

-->
```

* The only thing we did different so far was to save the results. The "I" (input) command will do another analysis using the file as an input. In this case, it means the signal passes through the circuit twice. Notice that the old data is displayed on the screen as it is read in, then the new data is written on the next line on the screen. The printer gets only the new data.

```
--> I
Old file name?  EX4
        freq        node   volts          dB   phase
        1800         3    .40453         -7.9 -114.4080
        1800         3    .163644       -15.7 +131.1840
        1900         3    .656753        -3.7 -132.1340
        1900         3    .431325        -7.3  +95.7316
        2000         3    .976801        -0.2 -176.1970
        2000         3    .95414         -0.4   +7.6055
        2100         3    .719885        -2.9 +137.3380
        2100         3    .518234        -5.7  -85.3244
        2200         3    .462486        -6.7 +118.1920
        2200         3    .213893       -13.4 -123.6160
-->
```

* This is the response going through two identical stages. Notice that the frequencies were taken from the file, rather than being specified in the command. The file does not have to be generated by the same circuit. "ID" would cause the new results to be saved in another file.
* Notice that the program always asks for "Old file name" or "New file name". If the command was "ID" it would have asked for both. The "old file" must already exist, and will simply be read. The "new file" will be created if it doesn't already exist, and it will always be written to.
* Comparisons work similarly. The files are compared instead of being hooked together. The "C" command will do a comparison.

```
--> C
Old file name?  EX4
        freq        node   volts          dB   phase
        1800         3    .40453         -7.9 -114.4080
        1800         3    .999999        +0.0   +0.0003
        1900         3    .656753        -3.7 -132.1340
        1900         3    1              +0.0   -0.0003
        2000         3    .976801        -0.2 -176.1970
        2000         3    .999999        +0.0   -0.0005
        2100         3    .719885        -2.9 +137.3380
        2100         3    .999999        +0.0   -0.0004
        2200         3    .462486        -6.7 +118.1920
        2200         3    .999999        +0.0   -0.0001
```

* This command works similarly to "I", but the final list shows all 1 volt, 0 dB, 0 degrees. This means that the circuits perform the same. The small error is due to computer rounding errors.
* What happens if the circuits are different. Let's change the input from 1 volt to 2 volts and invert its phase.

```
-->  M1
    1     V    0  0   1          -2
    1     V    0  0  -2
-->  C
Old file name?  EX4
         freq         node   volts          dB    phase
        1800            3   .40453         -7.9 -114.4080
        1800            3   2              +6.0 -180.0000
        1900            3   .656753        -3.7 -132.1340
        1900            3   2              +6.0 +180.0000
        2000            3   .976801        -0.2 -176.1970
        2000            3   2              +6.0 +180.0000
        2100            3   .719885        -2.9 +137.3380
        2100            3   2              +6.0 +180.0000
        2200            3   .462486        -6.7 +118.1920
        2200            3   2              +6.0 +180.0000
-->
```

* Exactly what you would expect. This only represents a sample of what these commands are capable of. By repeating "ID" commands, infinite circuits can be analyzed in stages, and the computer adds them together.

Example 5
Automatic operation

* The "F" command turns over command to a file. The file "EX5" will run all the examples, so far, without any intervention from the operator. This will create files, so be sure you can write on the disk.

```
--> F
Old file name? EX5
This file runs all the examples.
..
..
```

* If you want to pause to look at what it is doing, press control-s. Then press any key to continue. If you want to terminate this, press control-c.

Example 6
Branch data

* This example uses the same circuit as example 2, to demonstrate the branch data utility.
* First, load the circuit as you did before.

```
--> BD
Old file name?  EX2.CKT
Example 2  --  bandpass filter
    1    V    0  0   1
    2    R    1  0   100000        1.0
    3    R    1  0   470           1.0
    4    C    1  2   8.2E-09       1.0
    5    C    1  3   8.2E-09       1.0
    6    R    2  3   200000        1.0
    7    V    0  2   10000        25.0
    8    R    3  0   100          25.0
```

* The next step is to get the data for all nodes. This is done with the "A" command. We will get the data for d.c. (0 Hz) and 2000 Hz. (Two frequencies) Save the results in a file.

```
-->  AD 0,2000
New file name?  EX2.DAT
Heading ?
All nodes -- filter example
         freq         node   volts          dB     phase
        0              1   4.67801E-03
        0              2   0
        0              3   0
         freq         node   volts          dB     phase
        2000           1   .0474009      -26.5  -86.0791
        2000           2   9.77828E-05  -80.2   +4.3925
        2000           3   .976799       -0.2 -176.1970
         freq         node   volts          dB     phase
```

* This data will be used by the branch data utility. Notice that phase and dB are not shown for d.c.
* At this time, exit Circuit Analysis and run "BRANCH.BAS" from basic. (This requires Microsoft Basic)

```
-->  Q
A>MBASIC BRANCH
```

* At this time, Basic signs on, loads "BRANCH.BAS", and runs it. It will ask whether to send the output to the printer, and for the two files required. (Generated by ECA) The component file was generated by the BD command. The results file was generated by the AD command.

```
Branch data utility
For use with:
Electronic Circuit Analysis 1.0

Output to printer?  n
Component file name? EX2.CKT
Results file name? EX2.DAT
All nodes -- filter example
         freq        node    volts        dB      phase
        0             1    4.67801E-03
        0             2    0
        0             3    0
Example 2  --   bandpass filter
branch   type      volts             current          v * i
 1        V        1                 9.95322E-06      9.95322E-06
 2        R        .995322           9.95322E-06      9.90666E-06
 3        R        4.67801E-03       9.95322E-06      4.65612E-08
 4        C        4.67801E-03       0                0
 5        C        4.67801E-03       0                0
 6        R        0                 0                0
 7        REF      0
 7        V        0                 0                0
 8        R        0                 0                0
?
         freq        node    volts        dB      phase
        2000          1    .0474009    -26.5   -86.0791
        2000          2    9.77828E-05 -80.2    +4.3925
        2000          3    .976799      -0.2 -176.1970
Example 2  --   bandpass filter
branch   type      volts             current          v * i
 1        V        1                 9.97881E-06      9.97881E-06
 2        R        .997881           9.97881E-06      9.95767E-06
 3        R        .0474009          1.00853E-04      4.78052E-06
 4        C        .0474018          4.88448E-06      2.31533E-07
 5        C        .978047           1.00782E-04      9.85697E-05
 6        R        .976897           4.88448E-06      4.77164E-06
 7        REF      9.77828E-05
 7        V        .977828           1.0149E-04       9.92399E-05
 8        R        .010149           1.0149E-04       1.03002E-06
```

* The program displays the node data, as read from the file, then displays the data for each branch.
* Some new information is now available including separate data for sources and their resistors. In this example, branch 1 is the active part and branch 2 is its series resistor. The extra branch 7 with type "REF" is the input to the controlled source
* The column marked "v * i" shows power dissipation in resistors, and the v * i supplied by sources, which is power in a d.c. circuit or if there is a resistive load.
* After displaying data for the first frequency, the program displays "?" and waits for a "return". It then displays the data for next frequency. This repeats until all frequencies are displayed.